

**PEMBANGUNAN APLIKASI MONITORING DAN SELEKSI
ATLET (STUDI KASUS: CABANG OLAHRAGA BASKET, SEPAK
BOLA, DAN BOLA VOLI)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Helfi Pangestu
NIM: 145150201111105



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PEMBANGUNAN APLIKASI MONITORING DAN SELEKSI ATLET (STUDI KASUS:
CABANG OLAHRAGA BASKET, SEPAK BOLA, DAN BOLA VOLI)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Helfi Pangestu
NIM: 145150201111105

Skripsi ini telah diuji dan dinyatakan lulus pada
10 Oktober 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Bayu Priyambadha, S.Kom, M.Kom
NIP: 19820909 200812 1 000

Achmad Arwan, S.Kom, M.Kom
NIP: 19840815 200812 1 004

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Oktober 2018



Helfi Pangestu

NIM: 145150201111105

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “PEMBANGUNAN APLIKASI MONITORING DAN SELEKSI ATLET”.

Untuk kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Allah yang Maha Esa Karena atas kehendak dan nikmat-NYA laporan skripsi ini telah selesai dengan baik.
2. Ibu Maryati, Mukti Ageng Wicaksono dan Ndaru Agung Pamungkas atas nasehat, kasih sayang dan segala bentuk dukungan serta doanya.
3. Bayu Priyambadha, S.Kom, M.Kom selaku dosen pembimbing pertama yang telah memberikan support dan bimbingannya kepada ananda untuk segera menyelesaikan skripsi ini.
4. Achmad Arwan, S.Kom, M.Kom selaku dosen pembimbing kedua yang selalu membantu ananda dalam menuntaskan laporan skripsi.
5. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
6. Rania Akhmalia atas dukungannya, kesediaan meluangkan waktu dan tenaga juga doanya sampai pengerjaan skripsi selesai.
7. Teman-temanku tercinta di program studi teknik informatika angkatan 2014 yang selalu memberikan semangatnya sehingga penulis dapat menyelesaikan skripsi ini.
8. Dan orang-orang yang selalu mendukung dan mendoakan ananda yang tidak dapat ananda ucapkan satu persatu. Terimakasih atas semua doa dan dukungan baik inmateril maupun non meteril.

Penulis menyadari masih banyak kekurangan dalam penyusunan laporan PKL ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan PKL ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 10 Oktober 2018

Penulis

helfipangestu@gmail.com

ABSTRAK

Olahraga adalah aktivitas yang dapat meningkatkan kebugaran seseorang. Pemerintah Indonesia membentuk unit-unit yang berfungsi mewadahi keterampilan di bidang olahraga. Tidak hanya skala nasional, unit juga terbentuk di skala kampus. Selain mewadahi keterampilan dan menyalurkan bakat masyarakat, unit olahraga juga berfungsi menyaring calon atlet profesional. Untuk mendapatkan atlet profesional, setiap unit mengadakan pertandingan internal guna mengamati potensi calon atlet. Hal ini dianggap kurang objektif sebab keadaan seseorang tidak dapat diamati hanya pada satu atau dua kondisi namun pada kondisi yang berkelanjutan. Dalam proses pengamatan berkelanjutan atau monitoring atlet, pelatih cabang olahraga (cabor) kesulitan melakukan dokumentasi performa para atlet karena keterbatasan sumber daya. Dilatarbelakangi permasalahan diatas peneliti mengajukan sebuah sistem yang dapat memonitoring performa dan menyimpan hasil monitoring para atlet menggunakan metode Waterfall development

Dari implementasi yang telah dilakukan menghasilkan sistem monitoring dan seleksi atlet memiliki fitur tim manajemen, atlet achievement, riwayat cedera, dan rekomendasi atlet. Sistem ini dibangun dengan menggunakan arsitektural Mode-View-Controller (MVC) dengan menggunakan framework Laravel. Hasil pengujian *whitebox* berupa *basis path testing* dan *blackbox* yang dilakukan berupa validasi menghasilkan validitas sebesar 100%.

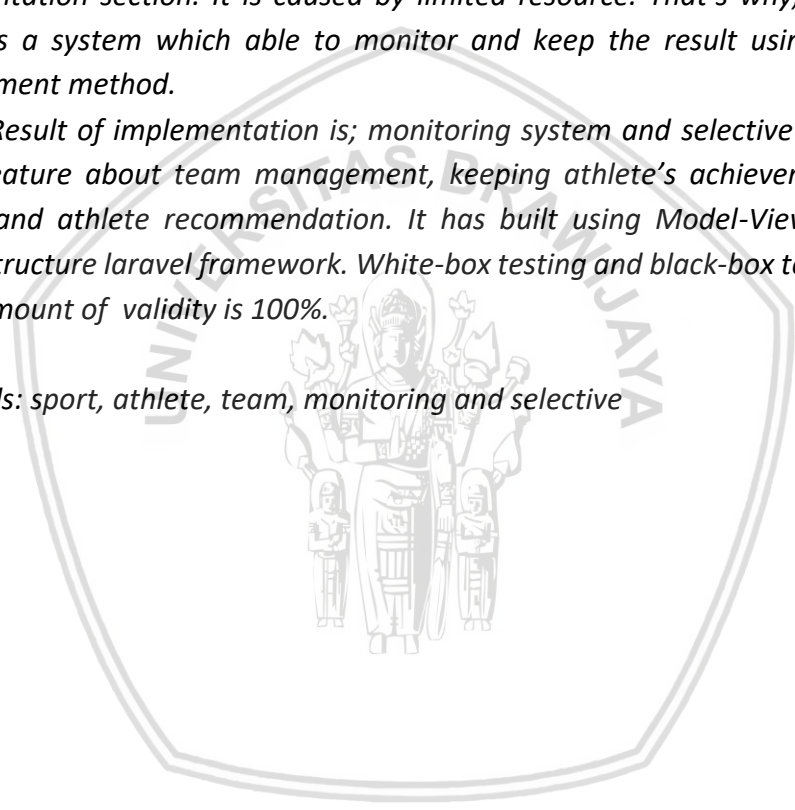
Kata kunci: olahraga, atlet, tim, monitor dan seleksi

Abstract

Sport is an activity which can improve someone's health. Indonesia government creates an organization that aims to accommodate people based on their hobbies especially in sport. Organization scale is not only national but also available in campus. Beside of accommodating people, the organization aims gathering people who has good potential. To select and monitor prospective athlete, organization management do internal sparing. But it is not objective because someone's condition can not only monitor by once or twice workout. In case continually monitoring, coach has faced trouble in performance documentation section. It is caused by limited resource. That's why, researcher proposes a system which able to monitor and keep the result using waterfall development method.

Result of implementation is; monitoring system and selective athlete has many feature about team management, keeping athlete's achievement, injury history and athlete recommendation. It has built using Model-View-Controller (MVC) structure laravel framework. White-box testing and black-box testing result in the amount of validity is 100%.

keywords: sport, athlete, team, monitoring and selective



DAFTAR ISI

PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vii
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xx
DAFTAR LAMPIRAN	xxi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Landasan Teori	5
2.2.1 Basket	5
2.2.2 Sepak Bola	6
2.2.3 Bola Voli	7
2.2.4 Pengembangan Perangkat Lunak.....	7
2.2.5 <i>Pseudocode</i>	15
2.2.6 Teknologi Pengembangan Perangkat Lunak	15
BAB 3 METODOLOGI	24
3.1 Studi Literatur	25
3.2 Analisis Kebutuhan	25
3.2.1 Studi Lapangan	25
3.3 Perancangan Sistem.....	26
3.4 Implementasi	27
3.5 Pengujian dan Analisis	27
3.6 Kesimpulan.....	28

BAB 4 ANALISIS KEBUTUHAN	29
4.1 Elisitasi Kebutuhan.....	29
4.1.1 Identifikasi Aktor	30
4.2 Kebutuhan Fungsional	31
4.3 Kebutuhan Non Fungsional.....	39
4.4 Use Case Diagram	40
4.5 Use Case Scenario	42
4.5.1 Login	42
4.5.2 Logout	42
4.5.3 Register	43
4.5.4 Reset Password	43
4.5.5 Melihat Daftar <i>Top</i> Tim	44
4.5.6 Melihat Daftar <i>Top</i> Atlet	44
4.5.7 Melihat Tampilan <i>Home</i>	44
4.5.8 Melihat Tampilan <i>Dashboard</i>	45
4.5.9 Melihat Profile Pelatih	45
4.5.10 Mengubah Profile Pelatih	45
4.5.11 Melihat Daftar <i>Achievement</i> Pelatih.....	46
4.5.12 Menambahkan <i>Achievement</i> Pelatih	46
4.5.13 Melihat Detail <i>Achievement</i> Pelatih.....	47
4.5.14 Mengubah <i>Achievement</i> Pelatih	47
4.5.15 Menghapus <i>Achievement</i> Pelatih	48
4.5.16 Melihat Detail Monitoring	48
4.5.17 Menambah Monitoring.....	48
4.5.18 Mengubah Score Atlet	49
4.5.19 Mengakhiri Monitoring Atlet	50
4.5.20 Melihat Daftar Rekomendasi Atlet	50
4.5.21 Melihat Daftar Atlet Hasil Monitoring Aktifitas Exercise.....	51
4.5.22 Melihat Daftar Atlet Hasil Monitoring Aktifitas Sparring.....	51
4.5.23 Melihat Daftar Atlet Hasil Monitoring Aktifitas Championship	51
4.5.24 Melihat Daftar Tim	52
4.5.25 Melihat Detail Tim.....	52

4.5.26 Menambahkan Tim	53
4.5.27 Mengubah Tim	53
4.5.28 Menghapus Tim.....	54
4.5.29 Melihat Daftar <i>Achievement</i> Tim	54
4.5.30 Melihat Detail <i>Achievement</i> Tim.....	55
4.5.31 Menambahkan <i>Achievement</i> Tim	55
4.5.32 Mengubah <i>Achievement</i> Tim	55
4.5.33 Menghapus <i>Achievement</i> Tim.....	56
4.5.34 Melihat Daftar Atlet	56
4.5.35 Melihat Detail Atlet.....	57
4.5.36 Menambahkan Atlet	57
4.5.37 Mengubah Atlet	58
4.5.38 Menghapus Atlet.....	58
4.5.39 Melihat Daftar <i>Achievement</i> Atlet	59
4.5.40 Melihat Detail <i>Achievement</i> Atlet.....	59
4.5.41 Menambahkan <i>Achievement</i> Atlet	59
4.5.42 Mengubah <i>Achievement</i> Atlet	60
4.5.43 Menghapus <i>Achievement</i> Atlet.....	60
4.5.44 Melihat Daftar Riwayat Riwayat Cedera Atlet	61
4.5.45 Melihat Detail Riwayat Cedera Atlet.....	61
4.5.46 Menambahkan Riwayat Cedera Atlet	61
4.5.47 Mengubah Riwayat Cedera Atlet	62
4.5.48 Menghapus Riwayat Cedera Atlet	62
4.5.49 Melihat Daftar Aktifitas Atlet.....	63
4.5.50 Melihat Detail Parameter Penilaian	63
4.5.51 Melihat Daftar Parameter Tetap Penilaian Atlet	64
4.5.52 Melihat Daftar Parameter Tidak Tetap Penilaian Atlet	64
4.5.53 Menambahkan Parameter Tidak Tetap Penilaian Atlet.....	64
4.5.54 Mengubah Parameter Tidak Tetap Penilaian Atlet.....	65
4.5.55 Menghapus Parameter Tidak Tetap Penilaian Atlet	66
4.5.56 Mengubah Persentase Tipe Parameter Penilaian.....	66
4.5.57 Melihat Daftar Cabang Olahraga	67

4.5.58 Melihat Detail Cabang Olahraga	67
4.5.59 Menambahkan Cabang Olahraga.....	68
4.5.60 Mengubah Cabang Olahraga.....	68
4.5.61 Menghapus Cabang Olahraga	69
4.5.62 Melihat Daftar Pelatih.....	69
4.5.63 Melihat Detail Pelatih	69
4.5.64 Menambahkan Pelatih	70
4.5.65 Mengubah pelatih	70
4.5.66 Menghapus Pelatih	71
BAB 5 PERANCANGAN DAN IMPLEMENTASI	72
5.1 Perancangan	72
5.1.1 Pemodelan <i>Sequence Diagram</i>	72
5.1.2 Pemodelan <i>Class Diagram</i>	74
5.1.3 Perancangan Algoritme.....	77
5.1.4 Perancangan Basis Data	80
5.1.5 Perancangan Antarmuka.....	81
5.2 Implementasi	83
5.2.1 Spesifikasi Sistem	84
5.2.2 Implementasi Basis Data	84
5.2.3 Implementasi Antarmuka	86
BAB 6 PENGUJIAN	88
6.1 Pengujian Unit.....	88
6.1.1 Pengujian Unit Algoritma Menambahkan Tim.....	88
6.1.2 Pengujian Unit Algoritma Melihat Detail Tim	91
6.1.3 Pengujian Unit Algoritma Mengubah Score Atlet.....	94
6.1.4 Hasil Kasus Uji	97
6.2 Pengujian Validasi	97
6.2.1 Pengujian <i>Login</i>	98
6.2.2 Pengujian <i>Logout</i>	99
6.2.3 Pengujian <i>Register</i>	100
6.2.4 Pengujian <i>Reset password</i>	101
6.2.5 Pengujian Melihat Daftar Top Tim	102

6.2.6 Pengujian Melihat Daftar Top Atlet	102
6.2.7 Pengujian Melihat Profile Pelatih.....	102
6.2.8 Pengujian Melihat Daftar <i>Achievement</i> Pelatih	103
6.2.9 Pengujian Menambahkan <i>Achievement</i> Pelatih	103
6.2.10 Pengujian Melihat Detail <i>Achievement</i> Pelatih.....	104
6.2.11 Pengujian Mengubah <i>Achievement</i> Pelatih	104
6.2.12 Pengujian Menghapus <i>Achievement</i> Pelatih	105
6.2.13 Pengujian Mengubah Profile Pelatih	106
6.2.14 Pengujian Melihat Detail Monitoring.....	107
6.2.15 Pengujian Menambahkan Monitoring	107
6.2.16 Pengujian Melihat daftar atlet hasil monitoring aktifitas exercise	108
6.2.17 Pengujian Melihat daftar atlet hasil monitoring aktifitas sparing	108
6.2.18 Pengujian Melihat daftar atlet hasil monitoring aktifitas championship.....	109
6.2.19 Pengujian Melihat Daftar Rekomendasi Atlet	109
6.2.20 Pengujian Melihat Daftar Tim	110
6.2.21 Pengujian Melihat Detail Tim.....	110
6.2.22 Pengujian Menambahkan Tim	111
6.2.23 Pengujian Mengubah Tim	112
6.2.24 Pengujian Menghapus Tim.....	112
6.2.25 Pengujian Melihat Daftar <i>Achievement</i> Tim	113
6.2.26 Pengujian Melihat Detail <i>Achievement</i> Tim.....	113
6.2.27 Pengujian Menambahkan <i>Achievement</i> Tim	114
6.2.28 Pengujian Mengubah <i>Achievement</i> Tim	114
6.2.29 Pengujian Menghapus <i>Achievement</i> Tim.....	115
6.2.30 Pengujian Melihat Daftar Atlet	116
6.2.31 Pengujian Melihat Detail Atlet	116
6.2.32 Pengujian Menambahkan Atlet	117
6.2.33 Pengujian Mengubah Atlet	117
6.2.34 Pengujian Menghapus Atlet.....	118
6.2.35 Pengujian Melihat Daftar <i>Achievement</i> Atlet	118

6.2.36 Pengujian Melihat Detail <i>Achievement</i> Atlet	119
6.2.37 Pengujian Menambahkan <i>Achievement</i> Atlet	119
6.2.38 Pengujian Mengubah <i>Achievement</i> Atlet	120
6.2.39 Pengujian Menghapus <i>Achievement</i> Atlet	120
6.2.40 Pengujian Melihat Daftar Riwayat Cedera Atlet	121
6.2.41 Pengujian Melihat Detail Riwayat Cedera Atlet	122
6.2.42 Pengujian Menambahkan Riwayat Cedera Atlet	122
6.2.43 Pengujian Mengubah Riwayat Cedera Atlet	123
6.2.44 Pengujian Menghapus Riwayat Cedera Atlet	123
6.2.45 Pengujian Melihat Daftar Aktifitas Atlet	124
6.2.46 Pengujian Melihat Detail Parameter Penilaian	124
6.2.47 Pengujian Melihat Parameter Tetap Penilaian Atlet	125
6.2.48 Pengujian Melihat Daftar Parameter Tidak Tetap Penilaian Atlet	125
6.2.49 Pengujian Menambahkan Parameter Tidak Tetap Penilaian Atlet	126
6.2.50 Pengujian Mengubah Parameter Tidak Tetap Penilaian Atlet	126
6.2.51 Pengujian Menghapus Parameter Tidak Tetap Penilaian Atlet	127
6.2.52 Pengujian Mengubah Persentase Tipe Parameter Penilaian..	128
6.2.53 Pengujian Melihat Daftar Pelatih	130
6.2.54 Pengujian Melihat Detail Pelatih	130
6.2.55 Pengujian Menambahkan Pelatih	130
6.2.56 Pengujian Mengubah Pelatih	131
6.2.57 Pengujian Menghapus Pelatih	131
6.2.58 Pengujian Melihat Daftar Cabang Olahraga	132
6.2.59 Pengujian Melihat Detail Cabang Olahraga	132
6.2.60 Pengujian Menambahkan Cabang Olahraga	133
6.2.61 Pengujian Mengubah Cabang Olahraga	133
6.2.62 Pengujian Menghapus Cabang Olahraga	134
6.2.63 Pengujian Melihat Tampilan <i>Home</i>	135
6.2.64 Pengujian Melihat Tampilan <i>Dashboard</i>	135
6.2.65 Pengujian Mengubah Score Atlet	135

6.2.66 Pengujian Mengakhiri Monitoring Atlet	137
6.2.67 Hasil Pengujian Validasi.....	138
6.3 Pengujian Compatibility.....	138
BAB 7 KESIMPULAN.....	140
7.1 Kesimpulan.....	140
7.2 Saran	140
DAFTAR PUSTAKA.....	141



DAFTAR TABEL

Tabel 2.1 Notasi <i>Use Case Diagram</i>	11
Tabel 2.2 Detail <i>Use Case Scenario</i>	12
Tabel 2.3 Notasi <i>Sequence Diagram</i>	13
Tabel 2.4 Notasi <i>Class Diagram</i>	14
Tabel 2.5 <i>Pseudocode</i>	15
Tabel 2.6 <i>Cyclometric Complexity</i>	22
Tabel 4.1 Table Identifikasi Aktor	31
Tabel 4.2 Definisi dan Spesifikasi Kebutuhan Fungsional Sistem	31
Tabel 4.3 Daftar Kebutuhan Non Fungsional	39
Tabel 4.4 <i>Use Case. Scenario Login</i>	42
Tabel 4.5 <i>Use Case Scenario Logout</i>	42
Tabel 4.6 <i>Use Case Scenario Register</i>	43
Tabel 4.7 <i>Use Case Scenario Reset Password</i>	43
Tabel 4.8 <i>Use Case. Scenario</i> Melihat Daftar Top Tim	44
Tabel 4.9 <i>Use Case Scenario</i> .Melihat Daftar Top Atlet	44
Tabel 4.10 <i>Use Case Scenario</i> Melihat Tampilan Home	44
Tabel 4.11 Melihat Tampilan <i>Dashboard</i>	45
Tabel 4.12 <i>Use Case Scenario</i> Melihat Profile Pelatih	45
Tabel 4.13 <i>Use Case Scenario</i> Mengubah Profile Pelatih	45
Tabel 4.14 <i>Use Case Scenario</i> Melihat Daftar Achievement Pelatih	46
Tabel 4.15 <i>Use Case Scenario</i> Menambah <i>Achivement</i> Pelatih	46
Tabel 4.16 Melihat Detail <i>Achievement</i> Pelatih	47
Tabel 4.17 <i>Use Case Scenario</i> Mengubah <i>Achievement</i> Pelatih	47
Tabel 4.18 <i>Use Case Scenario</i> Menghapus Achievement Pelatih	48
Tabel 4.19 <i>Use Case Scenario</i> Melihat Detail Monitoring	48
Tabel 4.20 <i>Use Case Scenario</i> Menambah Monitoring.....	48
Tabel 4.21 <i>Use Case Scenario</i> Mengubah Score Atlet	49
Tabel 4.22 <i>Use Case Scenario</i> Mengakhiri Monitoring Atlet	50
Tabel 4.23 <i>Use Case Scenario</i> Melihat Daftar Rekomendasi Atlet	50

Tabel 4.24 <i>Use Case Scenario</i> Melihat Daftar Atlet Hasil Monitoring Aktifitas Exercise	51
Tabel 4.25 <i>Use Case Scenario</i> Melihat Daftar Atlet Hasil Monitoring Aktifitas Sparing.....	51
Tabel 4.26 <i>Use Case Scenario</i> Melihat Daftar Atlet Hasil Monitoring Aktifitas Championship.....	52
Tabel 4.27 <i>Use Case Scenario</i> Melihat Daftar Tim	52
Tabel 4.28 <i>Use Case Scenario</i> Melihat Detail Tim.....	52
Tabel 4.29 <i>Use Case Scenario</i> Menambahkan Tim	53
Tabel 4.30 <i>Use Case Scenario</i> Mengubah Tim	53
Tabel 4.31 <i>Use Case Scenario</i> Menghapus Tim.....	54
Tabel 4.32 <i>Use Case Scenario</i> Melihat Daftar Achievement Tim	54
Tabel 4.33 <i>Use Case Scenario</i> Melihat Detail Achievement Tim.....	55
Tabel 4.34 <i>Use Case Scenario</i> Menambahkan Achievement Tim	55
Tabel 4.35 <i>Use Case Scenario</i> Mengubah Achievement Tim	55
Tabel 4.36 <i>Use Case Scenario</i> Menghapus Achievement Tim.....	56
Tabel 4.37 <i>Use Case Scenario</i> Melihat Daftar Atlet	56
Tabel 4.38 <i>Use Case Scenario</i> Melihat Detail Atlet.....	57
Tabel 4.39 <i>Use Case Scenario</i> Menambahkan Atlet	57
Tabel 4.40 <i>Use Case Scenario</i> Mengubah Atlet	58
Tabel 4.41 <i>Use Case Scenario</i> Menghapus Atlet.....	58
Tabel 4.42 <i>Use Case Scenario</i> Melihat Daftar Achievement Atlet	59
Tabel 4.43 <i>Use Case Scenario</i> Meliaht Detail Achievement Atlet.....	59
Tabel 4.44 <i>Use Case Scenario</i> Menambahkan Achievement Atlet	59
Tabel 4.45 <i>Use Case Scenario</i> Mengubah Achievement Atlet	60
Tabel 4.46 <i>Use Case Scenario</i> Menghapus Achievement Atlet.....	60
Tabel 4.47 <i>Use Case Scenario</i> Melihat Daftar Riwayat Cedera Atlet.....	61
Tabel 4.48 <i>Use Case Scenario</i> Melihat Detail Riwayat Cedera Atlet.....	61
Tabel 4.49 <i>Use Case Scenario</i> Menambahkan Riwayat Cedera Atlet	61
Tabel 4.50 <i>Use Case Scenario</i> Mengubah Riwayat Cedera Atlet	62
Tabel 4.51 <i>Use Case Scenario</i> Menghapus Riwayat Cedera Atlet	62
Tabel 4.52 <i>Use Case Scenario</i> Melihat Daftar Aktifitas Atlet	63
Tabel 4.53 <i>Use Case Scenario</i> Melihat Detail Parameter Penilaian.....	63

Tabel 4.54 <i>Use Case Scenario</i> Melihat Daftar Parameter Tetap Penilaian Atlet ..	64
Tabel 4.55 <i>Use Case Scenario</i> Melihat Daftar Parameter Tidak Tetap Penilaian Atlet	64
Tabel 4.56 <i>Use Case Scenario</i> Menambahkan Parameter Tidak Tetap Penilaian Atlet.....	64
Tabel 4.57 <i>Use Case Scenario</i> Mengubah Parameter Tidak Tetap Penilaian Atlet65	
Tabel 4.58 <i>Use Case Scenario</i> Menghapus Parameter Tidak Tetap Penilaian Atlet	66
Tabel 4.59 <i>Use Case Scenario</i> Mengubah Persentase Tipe Parameter Penilaian.	66
Tabel 4.60 <i>Use Case Scenario</i> Melihat Daftar Cabang Olahraga	67
Tabel 4.61 <i>Use Case Scenario</i> Meliaht Detail Cabang Olahraga	67
Tabel 4.62 <i>Use Case Scenario</i> Menambahkan Cabang Olahraga.....	68
Tabel 4.63 <i>Use Case Scenario</i> Mengubah Cabang Olahraga.....	68
Tabel 4.64 <i>Use Case Scenario</i> Menghapus Cabang Olahraga	69
Tabel 4.65 <i>Use Case Scenario</i> Melihat Daftar Pelatih	69
Tabel 4.66 <i>Use Case Scenario</i> Melihat Detail Pelatih.....	69
Tabel 4.67 <i>Use Case Scenario</i> Menambahkan Pelatih	70
Tabel 4.68 <i>Use Case Scenario</i> Mengubah Pelatih	70
Tabel 4.69 <i>Use Case Scenario</i> Menghapus Pelatih	71
Tabel 5.1 Penjelasan Rancangan Antarmuka Menambahkan Tim	81
Tabel 5.2 Penjelasan Rancangan Antarmuka Melihat Detail Tim.....	82
Tabel 5.3 Penjelasan Rancangan Antarmuka Mengubah Score Atlet	83
Tabel 5.4 Spesifikasi Perangkat Keras	84
Tabel 5.5 Spesifikasi Perangkat Lunak	84
Tabel 6.1 <i>Pseudocode Algoritme</i> Menambahkan Tim	88
Tabel 6.2 Kasus Uji Algoritme Menambahkan Tim	90
Tabel 6.3 <i>Pseudocode Algoritme</i> Melihat Detail Tim.....	91
Tabel 6.4 Kasus Uji Algoritme Melihat Detail Tim.....	94
Tabel 6.5 <i>Pseudocode Algoritme</i> Mengubah Score Atlet	94
Tabel 6.6 Kasus Uji Algoritme Mengubah Score Atlet	96
Tabel 6.7 Kasus Uji Berhasil Login	98
Tabel 6.8 Kasus Uji Login Alternatif 1	98
Tabel 6.9 Kasus Uji Login Alternatif 2	99

Tabel 6.10 Kasus Uji Berhasil Logout	99
Tabel 6.11 Kasus Uji Berhasil Register	100
Tabel 6.12 Kasus Uji Register Alternatif 1	100
Tabel 6.13 Kasus Uji Berhasil <i>Reset Password</i>	101
Tabel 6.14 Kasus Uji Reset Password Alternatif 1.....	101
Tabel 6.15 Kasus Uji Berhasil Melihat Daftar Top Atlet	102
Tabel 6.16 Kasus Uji Berhasil Melihat Daftar Top Atlet	102
Tabel 6.17 Kasus Uji Berhasil Melihat Profile Pelatih	102
Tabel 6.18 Kasus Uji Berhasil Melihat Daftar <i>Achievement</i> Pelatih.....	103
Tabel 6.19 Kasus Uji Berhasil Menambahkan <i>Achievement</i> Pelatih	103
Tabel 6.20 Kasus Uji Berhasil Melihat Detail <i>Achievement</i> Pelatih	104
Tabel 6.21 Kasus Uji Berhasil Mengubah <i>Achievement</i> Pelatih	104
Tabel 6.22 Kasus Uji Berhasil Menghapus <i>Achievement</i> Pelatih	105
Tabel 6.23 Kasus Uji Menghapus <i>Achievement</i> Pelatih Alternatif 1	105
Tabel 6.24 Kasus Uji Berhasil Mengubah Profile Pelatih	106
Tabel 6.25 Kasus Uji Mengubah Profile Pelatih Alternatif 1	106
Tabel 6.26 Kasus Uji Berhasil Melihat Detail Monitoring	107
Tabel 6.27 Kasus Uji Berhasil Menambahkan Monitoring.....	107
Tabel 6.28 Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas exercise	108
Tabel 6.29 Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas sparing	108
Tabel 6.30 Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas championship.....	109
Tabel 6.31 Kasus Uji Berhasil Melihat Daftar Rekomendasi Atlet	109
Tabel 6.32 Kasus Uji Berhasil Melihat Daftar Tim	110
Tabel 6.33 Kasus Uji Berhasil Melihat Detail Tim.....	110
Tabel 6.34 Kasus Uji Berhasil Melihat Detail Tim.....	110
Tabel 6.35 Kasus Uji Berhasil Menambahkan Tim	111
Tabel 6.36 Kasus Uji Berhasil Menambahkan Tim	111
Tabel 6.37 Kasus Uji Berhasil Mengubah Tim	112
Tabel 6.38 Kasus Uji Berhasil Menghapus Tim	112
Tabel 6.39 Kasus Uji Berhasil Melihat Daftar <i>Achievement</i> Tim	113

Tabel 6.40 Kasus Uji Berhasil Melihat Detail <i>Achievement</i> Tim.....	113
Tabel 6.41 Kasus Uji Berhasil Menambahkan <i>Achievement</i> Tim	114
Tabel 6.42 Kasus Uji Berhasil Mengubah <i>Achievement</i> Tim	114
Tabel 6.43 Kasus Uji Berhasil Menghapus <i>Achievement</i> Tim	115
Tabel 6.44 Kasus Uji Menghapus <i>Achievement</i> Tim Alternatif 1	115
Tabel 6.45 Kasus Uji Berhasil Melihat Daftar Atlet	116
Tabel 6.46 Kasus Uji Berhasil Melihat Detail Atlet.....	116
Tabel 6.47 Kasus Uji Berhasil Menambahkan Atlet	117
Tabel 6.48 Kasus Uji Berhasil Mengubah Atlet	117
Tabel 6.49 Kasus Uji Berhasil Menghapus Atlet.....	118
Tabel 6.50 Kasus Uji Berhasil Melihat Daftar <i>Achievement</i> Tim	118
Tabel 6.51 Kasus Uji Berhasil Melihat Detail <i>Achievement</i> Atlet.....	119
Tabel 6.52 Kasus Uji Berhasil Menambahkan <i>Achievement</i> Atlet	119
Tabel 6.53 Kasus Uji Berhasil Mengubah <i>Achievement</i> Atlet	120
Tabel 6.54 Kasus Uji Berhasil Menghapus <i>Achievement</i> Atlet.....	120
Tabel 6.55 Kasus Uji Menghapus <i>Achievement</i> Atlet Alternatif 1	121
Tabel 6.56 Kasus Uji Berhasil Melihat Daftar Riwayat Cedera Atlet.....	121
Tabel 6.57 Kasus Uji Berhasil Melihat Detail Riwayat Cedera Atlet	122
Tabel 6.58 Kasus Uji Berhasil Menambahkan Riwayat Cedera Atlet	122
Tabel 6.59 Kasus Uji Berhasil Mengubah Riwayat Cedera Atlet.....	123
Tabel 6.60 Kasus Uji Berhasil Menghapus Riwayat Cedera Atlet	123
Tabel 6.61 Kasus Uji Berhasil Melihat Daftar Aktifitas Atlet.....	124
Tabel 6.62 Kasus Uji Berhasil Melihat Detail Parameter Penilaian.....	124
Tabel 6.63 Kasus Uji Berhasil Melihat Detail Parameter Penilaian.....	125
Tabel 6.64 Kasus Uji Berhasil Melihat Daftar Parameter Tidak Tetap Penilaian Atlet	125
Tabel 6.65 Kasus Uji Berhasil Menambahkan Parameter Tidak Tetap Penilaian Atlet	126
Tabel 6.66 Kasus Uji Berhasil Mengubah Parameter Tidak tetap Penilaian Atlet	127
Tabel 6.67 Kasus Uji Berhasil Menghapus Parameter Tidak Tetap Penilaian Atlet	127

Tabel 6.68 Kasus Uji Menghapus Parameter Tidak Tetap Penilaian Atlet Alternatif 1.....	128
Tabel 6.69 Kasus Uji Berhasil Mengubah Persentase Tipe Parameter Penilaian	128
Tabel 6.70 Kasus Uji Mengubah Persentase Tipe Parameter Penilaian Alternatif 1	129
Tabel 6.71 Kasus Uji Berhasil Melihat Daftar Pelatih.....	130
Tabel 6.72 Kasus Uji Berhasil Melihat Detail Pelatih	130
Tabel 6.73 Kasus Uji Berhasil Menambahkan Pelatih	130
Tabel 6.74 Kasus Uji Berhasil Mengubah Pelatih.....	131
Tabel 6.75 Kasus Uji Berhasil Menghapus Pelatih	132
Tabel 6.76 Kasus Uji Berhasil Melihat Daftar Cabang Olahraga	132
Tabel 6.77 Kasus Uji Berhasil Melihat Detail Cabang Olahraga	132
Tabel 6.78 Kasus Uji Berhasil Menambahkan Pelatih.....	133
Tabel 6.79 Kasus Uji Berhasil Mengubah Cabang Olahraga	134
Tabel 6.80 Kasus Uji Berhasil Menghapus Cabang Olahraga.....	134
Tabel 6.81 Kasus Uji Berhasil Melihat Tampilan <i>Home</i>	135
Tabel 6.82 Kasus Uji Berhasil Melihat Tampilan <i>Dashboard</i>	135
Tabel 6.83 Kasus Uji Berhasil Mengubah Score Atlet	135
Tabel 6.84 Kasus Uji Mengubah Score Atlet Alternatif 1	136
Tabel 6.85 Kasus Uji Mengubah Score Atlet Alternatif 2	136
Tabel 6.86 Kasus Uji Berhasil Mengakhiri Monitoring Atlet	137
Tabel 6.87 Kasus Uji Mengakhiri Monitoring Atlet Alternatif 1.....	137

DAFTAR GAMBAR

Gambar 2.1 Diagram Model Waterfall.....	8
Gambar 2.2 Ilustrasi Laravel Framework	18
Gambar 2.3 Notasi <i>Flowgraph</i>	22
Gambar 3.1 Diagram Alir Metode Penelitian.....	24
Gambar 4.1 Proses Bisnis Seleksi Tim Inti.....	30
Gambar 4.2 <i>Use Case Diagram</i> Sistem.....	41
Gambar 5.1 <i>Sequence Diagram</i> Menambahkan Tim	73
Gambar 5.2 <i>Sequence Diagram</i> Melihat Detail Tim.....	73
Gambar 5.3 <i>Sequence Diagram</i> Mengubah Score Atlet	74
Gambar 5.4 <i>Class Diagram</i> Perancangan Umum.....	75
Gambar 5.5 <i>Class Diagram</i> pada Model	76
Gambar 5.6 <i>Class Diagram</i> Controller	76
Gambar 5.7 <i>Entity Relational Diagram</i>	80
Gambar 5.8 Rancangan Antarmuka Menambahkan Tim.....	81
Gambar 5.9 Rancangan Antarmuka Melihat Detail Tim	82
Gambar 5.10 Rancangan Antarmuka Mengubah Score Atlet.....	83
Gambar 5.11 Hasil Implementasi Basis Data	85
Gambar 5.12 Tampilan Antarmuka Menambahkan Tim	86
Gambar 5.13 Tampilan Antarmuka Melihat Detail Tim	86
Gambar 5.14 Tampilan Antarmuka Mengubah Score Atlet	87
Gambar 6.1 <i>Flow Graph</i> Menambahkan Tim.....	89
Gambar 6.2 <i>Flow graph</i> Melihat Detail Tim.....	93
Gambar 6.3 <i>Flow graph</i> Mengubah Score Atlet	96
Gambar 6.4 Pengujian <i>Compatibility</i> pada <i>Opera Browser</i>	138
Gambar 6.5 Pengujian <i>Compatibility</i> pada <i>Microsoft Edge Browser</i>	139
Gambar 6.6 Pengujian <i>Compatibility</i> pada <i>Chrome Browser</i>	139

BAB 1 PENDAHULUAN

1.1 Latar belakang

Olahraga adalah salah satu aktivitas yang digunakan sebagai sarana meningkatkan kebugaran seseorang, tidak hanya jasmani tetapi juga rohani. Ada banyak jenis olahraga yang dapat dilakukan seseorang untuk meningkatkan kebugaran tubuh dengan menyesuaikan kemampuan dan bakat yang dimiliki. Pemerintah Indonesia membentuk unit-unit yang berfungsi mewadahi keterampilan di bidang olahraga dengan sarana dan prasarana yang cukup mewadahi. Unit ini terbentuk dari level terbesar dengan skala nasional hingga lingkup terkecil yaitu skala sekolah atau biasa dikenal dengan kelompok ekstrakurikuler. Sebagai contoh unit olahraga dengan skala nasional di bidang olahraga cabang sepak bola adalah PSSI atau Persatuan Sepak bola Seluruh Indonesia contoh lainnya pada cabang olahraga basket terdapat PERBASI atau Persatuan Basket Seluruh Indonesia. Sedangkan contoh unit terkecil berupa kelompok ekstra-kurikuler yang dapat ditemukan sejak tingkatan sekolah dasar.

Selain berfungsi untuk mewadahi keterampilan dan menyalurkan bakat penduduk Indonesia. Setiap unit juga bertugas mengumpulkan calon atlet yang berpotensi menjadi atlet profesional dan berlaga di sebuah kejuaraan. Untuk mendapatkan calon atlet profesional, setiap unit cabang olahraga mengadakan pertandingan guna mengamati potensi calon atlet. Sayangnya hal ini dianggap kurang objektif sebab keadaan seseorang tidak dapat diamati hanya pada satu atau dua kondisi tertentu namun pada kondisi yang berkelanjutan. Pernyataan tersebut disampaikan oleh salah satu ketua umum unit cabang olahraga sepeda atau dikenal dengan Pengurus Besar Ikatan Sport Sepeda Indonesia, Raja Sapta Oktohari (Mercy Raya, 2017).

Proses seleksi yang sama juga dilakukan di lingkup kecil yaitu unit kegiatan mahasiswa (UKM) di Universitas Brawijaya. Dalam wawancara bersama pelatih UKM olahraga basket pada jumat 20 Januari 2018, Andi Cahyono mengungkapkan bahwa ia seringkali mengalami kesulitan untuk menyeleksi atlet-atlet lokal untuk diikuti sertakan dalam kejuaraan. Kesulitan yang dialami secara teknis berupa pencatatan performa atlet selama 2-6 bulan terakhir. Pencatatan tersebut berguna untuk proses seleksi dan evaluasi atlet juga menjadi pendukung keputusan dalam menentukan delegasi kejuaraan.

Permasalahan serupa juga diungkapkan oleh mantan ketua Badan Internal Olahraga dan Seni (BIOS) unit kegiatan mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya, Vatikan Aulia Makkah. Vatikan menyatakan, saat ini seleksi atlet dilakukan berdasarkan hasil monitoring selama latihan. Setiap pelatih melakukan pelaporan hasil monitoring atlet kepada ketua divisi masing-masing cabang olahraga. Hasil monitoring

selama latihan pun tidak tercatat dan disimpan secara sistematis. Sebagai bahan pertimbangan lain masing-masing pelatih cabang olahraga juga mengadakan sparing atau latihan tanding internal tim. Menurut Vatikan, pencatatan performa yang tidak teratur akan menghambat proses seleksi. Sebab performa atlet yang diingat oleh pelatih hanya dalam kurun waktu singkat. Selaku ketua BIOS, ia juga tidak menutup kemungkinan beberapa pelatih cabang olahraga seperti pelatih voli, futsal dan basket yang sudah mulai teratur dan mencatat hasil monitoring ke dalam selembar kertas. Dikarenakan ketiga cabang olahraga tersebut paling banyak jumlah atletnya. Walaupun begitu, permasalahan baru bermunculan seperti dokumentasi yang tidak lengkap karena lembar kertas hasil monitoring selama latihan tersebut tidak tersimpan dengan baik.

Permasalahan-permasalahan diatas seringkali terjadi pada klub olahraga dengan lingkup kecil. keterbatasan sumber daya menjadi alasan proses monitoring atlet terhambat sehingga mempengaruhi hasil seleksi. Oleh sebab itu dibutuhkan sebuah sistem monitoring atlet yang dapat diakses dengan mudah dan juga menyajikan tampilan user-friendly bagi para pelatih yang terbiasa dengan pencatatan mode konvensional.

Metode yang akan digunakan dalam pembuatan sistem tersebut adalah Waterfall development. Waterfall development merupakan sebuah daur hidup pengembangan software yang melibatkan stake holder ikut dalam proses pengembangan produk secara berkala. Kelebihan metode waterfall dengan metode yang lainnya seperti waterfall adalah kecepatan waktu pengerjaan sistem dan membangun komunikasi yang konsisten bagi semua pihak yang terlibat sehingga membuat keberhasilan suatu proyek dapat terealisasi (Azdy & SN, 2012). Dengan adanya penelitian ini diharapkan proses seleksi calon atlet dari skala besar seperti nasional maupun skala terkecil seperti tingkat ekstrakurikuler di perguruan tinggi menjadi lebih objektif dan transparan. Sehingga dapat meningkatkan prestasi Indonesia di dunia olahraga.

Dilatarbelakangi permasalahan diatas peneliti mengajukan sebuah sistem yang dapat memonitoring performa para atlet pada saat latihan sehari-hari. Dengan memanfaatkan perkembangan teknologi dan jaringan, setiap pelatih dari unit cabang olahraga apapun baik skala kecil berkesempatan mendaftarkan pemain yang ia latih melalui sebuah website. Tidak hanya mendaftarkan data personal pemain seperti tinggi dan berat badan, pelatih juga dapat melaporkan performa setiap pemain pada setiap kali waktu latihan dengan parameter tertentu. Dengan begitu, setiap calon atlet memiliki rekam jejak performa yang dapat dimonitoring dalam kurun waktu tertentu oleh pelatih. Selain halaman untuk pelatih, sistem ini juga dilengkapi dashboard admin. Pada dashboard admin, seorang administrator dapat memantau keseluruhan performa atlet yang telah didaftarkan oleh masing-masing pelatihnya.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dikemukakan, rumusan masalah pada penelitian ini antara lain:

1. Bagaimana hasil analisis kebutuhan aplikasi monitoring dan seleksi atlet menggunakan metode waterfall software life cycle?
2. Bagaimana hasil rancangan aplikasi monitoring dan seleksi atlet yang sesuai dengan kebutuhan pengguna?
3. Bagaimana hasil implementasi aplikasi monitoring dan seleksi atlet menggunakan metode waterfall software life cycle?
4. Bagaimana hasil pengujian aplikasi monitoring dan seleksi atlet menggunakan metode waterfall software life cycle?

1.3 Tujuan

Tujuan dilakukannya penelitian ini antara lain:

1. Untuk merancang suatu sistem yang dapat digunakan sebagai pendukung keputusan dalam analisis keputusan penetapan atlet terbaik dalam suatu permainan Sepak Bola, bola voli, dan bola basket di Universitas Brawijaya.
2. Mempermudah para pelatih untuk menentukan siapa atlet terbaik berdasarkan kriteria dan alternatif yang diberikan sehingga pelatih dapat dengan mudah memilih sesuai kemampuan masing-masing atlet.

1.4 Manfaat

Manfaat dilakukan penelitian ini antara lain:

1. Untuk evaluasi pelatih terhadap pemain agar dapat meningkatkan performa pemain dan tim.
2. Untuk mendapatkan pemain terbaik dengan melakukan seleksi yang transparan.

1.5 Batasan masalah

Agar pembahasan penelitian tidak meluas keluar ranah penulis, maka ditentukan batasan masalah sebagai berikut:

1. Penelitian ini mengambil studi kasus cabang olahraga Sepak Bola, bola voli dan basket dengan skala Universitas Brawijaya.
2. Pembangunan aplikasi monitoring dan seleksi atlet berupa website.
3. Pengguna sistem ini adalah pelatih tim setiap cabang olahraga dan seorang administrator.

1.6 Sistematika pembahasan

Uraian singkat mengenai metodologi penelitian pada masing-masing bab adalah sebagai berikut:

BAB I Pendahuluan

Menjelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari “Pembangunan Aplikasi Monitoring dan Seleksi Atlet”

BAB II Landasan Kepustakaan

Membahas mengenai landasan teori yang terkait dengan penelitian. Pada bab ini juga dijelaskan terkait penelitian sejenis yang pernah dilakukan.

BAB III Metode Penelitian

Menjelaskan tentang langkah kerja yang dilakukan dalam penulisan diantaranya studi literatur, survey sistem, analisis sistem, desain sistem, implementasi sistem, dan perawatan sistem.

BAB IV Rekayasa Kebutuhan

Bab ini menjelaskan secara rinci yang terkait meliputi deskripsi umum dari sistem, rekayasa kebutuhan antar-muka sistem, kebutuhan perangkat keras dan lunak, kebutuhan fungsional, kebutuhan komunikasi, kebutuhan performansi, batasan desain sistem dan alur kerja sistem.

BAB V Perancangan dan Implementasi

Bab ini menjelaskan perancangan sistem penelitian serta implementasi sistem penelitian berupa implementasi perangkat keras, implementasi perangkat lunak, implementasi perancangan database dan implementasi aplikasi web sebagai *output* dari sistem.

BAB VI Pengujian dan Analisa

Bab ini membahas scenario pengujian, hasil pengujian beserta analisis hasil pengujian.

BAB VII Penutup

Bab ini membahas tentang kesimpulan dan saran yang diperoleh dari rumusan masalah penelitian dengan melakukan analisis dan pengujian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Penulis mengkaji beberapa penelitian terkait penelitian yang relevan dengan beberapa penelitian sebelumnya. Terdapat 3 penelitian yang berhubungan dengan sistem yang akan dibangun, yakni penelitian mengenai Prinsip Pelatihan Fisik Olahraga, penelitian mengenai Sistem Monitoring Peralatan Bengkel menggunakan Metode Waterfall, dan penelitian mengenai *Design and implementation of web based on Laravel framework*.

Dalam penelitian yang ditulis oleh Nala yang berjudul “Prinsip Pelatihan Fisik Olahraga” menjelaskan bahwa untuk mendapatkan penampilan atlet yang optimal diperlukan pelatihan yang dilakukan secara sistematis dengan pembobotan yang meningkat secara *progressive*. Seorang pelatih harus mampu memberikan sebuah pembobotan dan pencatatan pencapaian fisik atlet secara *progressive* agar atlet tidak mengalami cedera dan kelebihan beban terhadap fisiknya (Nala, 2011).

Penelitian selanjutnya yang dilakukan oleh Andriani dan Gea Siyoperman (Andriyani & Siyoperman, 2016) mengenai Sistem Monitoring Peralatan Bengkel menggunakan Metode Waterfall Dengan MVC Codeigniter. Dalam pengembangannya mengikuti Model SDLC Waterfall dengan menggunakan pendekatan Terstruktur. Metode waterfall digunakan pada Sistem Monitoring Peralatan Bengkel karena kemudahan dalam pengembangannya dan lingkup proyek yang tergolong kecil.

Kemudian penelitian yang dilakukan oleh He Ren Yu yang berjudul “*Design and implementation of web based on Laravel framework*” melakukan eksperimen pembangunan website untuk sebuah bisnis dengan waktu yang singkat namun berkualitas menggunakan *framework* laravel. Penelitian tersebut menghasilkan eksperimental dan simulasi terbukti bahwa desain web berdasarkan kerangka kerja Laravel, memiliki skalabilitas yang kuat, sehingga dapat meningkatkan efisiensi untuk berkembang (Yu, 2014).

2.2 Landasan Teori

2.2.1 Basket

Menurut Faizin (2015), basket adalah permainan olahraga yang dimainkan oleh dua regu, setiap regu memiliki jumlah pemain 12 orang dengan posisi 1 orang kapten, 5 orang pemain inti di lapangan, dan 7 orang pemain cadangan. Konsep permainan basket yaitu setiap regu berlomba mengumpulkan nilai sebanyak-banyaknya dengan cara memasukkan bola ke dalam keranjang. Basket dikategorikan dalam cabang olahraga bola besar yang bersifat kompetitif karena

tempo permainan cenderung lebih cepat. Olahraga ini dapat dilakukan didalam ruangan (*indoor*) maupun diluar ruangan (*outdoor*) (Faizin, 2015).

Basket memiliki 6 teknik permainan yang dapat diukur secara subjektif oleh pelatih. Teknik permainan tersebut antara lain:

1. Ball Handling atau kemampuan mengontrol bola.
2. Dribbling atau kemampuan menggiring bola.
3. Passing atau melempar/mengoper bola.
4. Shooting atau melakukan tembakan/lemparan bola ke dalam keranjang.
5. Offence atau penyerangan.
6. Deffense atau pertahanan.

2.2.2 Sepak Bola

Menurut Sucipto (2000), sepak bola adalah cabang olahraga permainan yang dimainkan oleh dua regu, masing-masing regu terdiri dari 10 orang pemain dan 1 orang pemain sebagai penjaga gawang. Konsep permainan sepak bola yaitu setiap regu belomba untuk mengumpulkan nilai sebanyak-banyaknya dengan cara memasukan bola ke dalam gawang lawan dan permainan tidak diperbolehkan menggunakan tangan serta lengan kecuali seorang penjaga gawang (Sucipto, (2000).).

Sepak bola memiliki 10 teknik permainan yang dapat diukur secara subjektif oleh pelatih (Herwin, 2004). Teknik permainan tersebut antara lain:

1. Pengenalan bola dengan bagian tubuh (*ball feeling*).
2. Mendang bola (*shooting*).
3. Mengoper bola pendek dan panjang atau melambung, menendang bola kegawang (*passing*).
4. Menggiring bola (*dribbling*).
5. Menghadapi lawan dan daerah bebas, menerima dan menguasai bola (*receiving and controlling the ball*).
6. Menyundul bola (*heading*) untuk bola lambung atau bola atas.
7. Gerak tipu (*feinting*) untuk melewati lawan.
8. Merebut bola (*tackling*) saat lawan menguasai bola
9. Melempar bola (*throw-in*) bila bola keluar lapangan untuk menghidupkan kembali permainan.
10. Teknik menjaga gawang (*goal keeping*).

2.2.3 Bola Voli

Menurut Julaintine & Yudiana (2010), voli adalah permainan memantulkan bola oleh tangan atau lengan dari dua regu yang bermain diatas lapangan yang mempunyai ukuran tertentu. Setiap regu berjumlah 10 orang dengan posisi 1 orang kapten, 5 orang pemain inti di lapangan, dan 4 orang pemain cadangan. Konsep permainan bola voli setiap regu berlomba mengumpulkan nilai sebanyak-banyaknya menyebrangkan bola melewati atas jaring kearah petak lawan yang disesuaikan dengan peraturan permainan. Olahraga voli dapat dilakukan didalam ruangan (indoor) maupun diluar ruangan (outdoor) (Juliantine & Yudiana, 2010).

Voli memiliki 5 teknik permainan yang dapat diukur secara subjektif oleh pelatih. Teknik permainan tersebut antara lain:

1. Servis.
2. Umpan (*Set-Up*).
3. Smash (*Spike*).
4. Block.
5. Libero.
6. Passing.

2.2.4 Pengembangan Perangkat Lunak

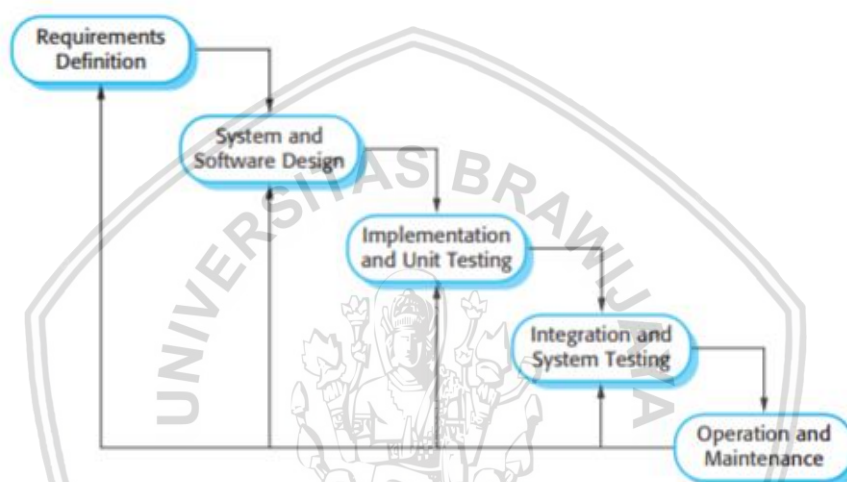
Menurut Sommerville (2011), rekayasa perangkat lunak merupakan runtutan kegiatan yang dilakukan untuk membangun dan mengembangkan perangkat lunak. Beberapa rekayasa perangkat lunak memiliki proses yang berbeda, namun semua harus mempunyai empat cakupan yang penting dilakukan, antara lain:

1. *Software specification*. Definisi dan batasan fungsionalitas wajib didefinisikan
2. *Software design and implementation*. Proses pembangunan dan pengembangan perangkat disesuaikan dengan spesifikasi yang telah disepakati.
3. *Software validation*. Validitas yang dimiliki sebuah perangkat harus dipastikan bernilai baik agar *user experience* juga bernilai baik.
4. *Software evolution*. Hasil pengembangan dapat dengan mudah dilakukan *scaling* oleh pengembang selanjutnya.

Model yang banyak diimplementasikan pada proses rekayasa perangkat lunak seperti; model Waterfall, model Prototyping, model Iterative, model Spiral, V-Model, dan model Big Bang. Pada penelitian ini, model yang diterapkan adalah model Waterfall.

2.2.4.1 Model Waterfall

Model Waterfall adalah proses pengembangan perangkat lunak sekuensial dimana fase pengembangan dijalankan dari atas menurun kebawah seperti air terjun dan mengharuskan menyelesaikan fase satu demi satu sebelum pindah ketahap berikutnya (Bassil, 2012). Awalnya, model Waterfall diusulkan oleh Winston W. Royce pada tahun 1970 untuk menjelaskan kemungkinan praktik rekayasa perangkat lunak (Royce, 1970). Model Waterfall mendefinisikan beberapa fase berturut-turut yang harus diselesaikan satu demi satu dan pindah ke tahap berikutnya hanya jika fase sebelumnya benar-benar selesai. Untuk alasan ini, model Waterfall rekursif dalam setiap fase dapat tanpa henti diulang sampai disempurnakan. Gambar 2.1 menggambarkan berbagai fase dari model waterfall.



Gambar 2.1 Diagram Model Waterfall

Sumber: Sommerville (2011)

Intinya, model waterfall memiliki 5 rangkaian penting yaitu; *Requirement Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operation and Maintenance*.

Tahap *Requirement Definition*: Rangkaian ini melakukan analisis terhadap kebutuhan perangkat lunak dan mengadakan pengumpulan data dengan melakukan pertemuan dengan customer, maupun mengumpulkan data-data tambahan baik yang di jurnal, artikel, maupun dari internet.

Tahap *System and Software Design*: Rangkaian ini melibatkan pengembang perangkat lunak dan perancang untuk menentukan rencana agar didapatkan solusi yang mencakup perancangan algoritma, perancangan arsitektur perangkat lunak, skema konseptual basis data dan desain diagram logis, desain konsep, desain antarmuka pengguna grafis, dan definisi struktur data.

Tahap *Implementation and Unit Testing*: Rangkaian ini mengacu pada implementasi persyaratan bisnis dan spesifikasi desain ke dalam program perangkat lunak, database, situs web, atau perangkat lunak yang dapat diprogram secara nyata melalui pemrograman dan penerapan. Tahap ini adalah dimana kode

sebenarnya ditulis dan disusun menjadi aplikasi operasional, dan dimana database dan file teks dibuat. Dengan kata lain, proses pengubahan keseluruhan persyaratan dan cetak biru menjadi lingkungan produksi.

Tahap *Integration and System Testing*: Ini juga dikenal sebagai verifikasi dan validasi yang merupakan proses untuk memeriksa apakah solusi perangkat lunak memenuhi persyaratan dan spesifikasi asli dan menyelesaikannya sesuai tujuannya. Sebenarnya, verifikasi adalah proses evaluasi perangkat lunak untuk menentukan apakah produk dari tahap pengembangan tertentu memenuhi persyaratan yang diberlakukan pada awal fase tersebut; Sementara, validasi adalah proses evaluasi perangkat lunak selama atau pada akhir proses pengembangan untuk menentukan apakah memenuhi persyaratan yang ditentukan (IEEE-STD-610, 1991). Selain itu, tahap pengujian adalah outlet untuk melakukan debugging di mana bug dan gangguan sistem ditemukan, diperbaiki, dan disempurnakan sesuai dengan bug yang ditemukan.

Tahap *Operation and Maintenance*: Ini adalah proses memodifikasi solusi perangkat lunak setelah memperbaiki keluaran, memperbaiki kesalahan, dan meningkatkan kinerja dan kualitas. Kegiatan *maintenance* tambahan dapat dilakukan pada tahap ini termasuk mengadaptasi perangkat lunak ke lingkungannya, mengakomodasi kebutuhan pengguna baru, dan meningkatkan keandalan perangkat lunak (Stellman & Greene, 2005).

Setiap perangkat lunak yang dikembangkan berbeda dan membutuhkan pendekatan SDLC yang berbeda pula. Beberapa situasi di mana penggunaan model Waterfall adalah yang paling tepat (Saxena & Upadhyay, 2016):

- a) Kebutuhan sistem didokumentasikan dengan sangat baik, jelas, dan tetap.
- b) Definisi kebutuhan stabil dan tidak ada perubahan definisi pada saat pengembangan.
- c) Teknologi dipahami dan tidak dinamis.
- d) Tidak ada kebutuhan sistem yang ambigu.
- e) Tersedianya sumber daya dengan keahlian yang dibutuhkan.

Kelebihan dari model Waterfall :

Keuntungan dari model Waterfall adalah proses pengembangan yang mudah dikontrol sebab jadwal yang *flexibel* sehingga dapat diatur untuk setiap tahap pengembangan serta produk dapat dengan mudah dilanjutkan setiap fase satu per satu.

Kekurangan dari model Waterfall :

Kerugian dari model Waterfall adalah proses revisi yang lebih sulit dibanding dengan model lainnya seperti scrum. Karena jika dokumentasi tahapan selanjutnya tidak tercatat dengan lengkap maka saat dibutuhkan kembali ke tahapan sebelumnya, akan menyulitkan pengembang.

2.2.4.2 Pendekatan Pengembangan Perangkat Lunak

Pendekatan pengembangan merupakan langkah yang dilakukan pada saat mengembangkan sebuah perangkat lunak. Beberapa bahasa pemrograman mendukung adanya pendekatan pengembangan ini. Namun tidak semua bahasa pemrograman yang menerapkan metode pendekatan ini (Toal, 2016). Terdapat 2 pendekatan pengembangan sistem, yakni:

1. Pendekatan Terstruktur

Pendekatan Terstruktur merupakan metode pemrograman dengan memperhatikan perintah dalam suatu *method* secara urut, sistematis, logis, dan terorganisir berdasarkan algoritma sederhana. Prinsip pendekatan ini dengan cara apabila suatu perintah dalam bentuk *code* telah dilakukan pada suatu proses, maka perintah selanjutnya tidak dapat kembali lagi ke baris sebelumnya, kecuali perintah perulangan/*looping*.

Berikut merupakan kelebihan dan kekurangan dari pendekatan Terstruktur (Naista, 2016):

Kelebihan:

- a. Penyelesaian masalah dengan skala kecil akan lebih efektif.
- b. *Source code* lebih sistematis dan mudah dipahami dengan algoritma yang juga sederhana.

Kekurangan:

- a. Masalah yang kompleks akan lebih sulit untuk ditemukan perintah penyelesaiannya.

2. Pendekatan *Object Oriented*

Pendekatan *Object Oriented* atau dalam bahasa Indonesia lebih dikenal dengan Orientasi Objek (OO) adalah pendekatan yang melihat suatu objek dalam pemrograman sebagai sekumpulan informasi (Satzinger, et al., 2007). Prinsip dari pendekatan ini adalah antara proses dan program tidak ada pemisahan. Selain itu antara entitas data juga tidak ada pemisahan. Operasi sistem dalam suatu *method* terdiri dari objek-objek. Sebuah objek dapat memberikan respon sesuai dengan pesan yang dikirim. Pendekatan OO dapat dibagi menjadi dua tahap utama yaitu *Object oriented analysis* (OOA) dan *Object oriented design* (OOD). OOA menempatkan sebuah objek pada domain permasalahan. Pada tahap ini, objek diidentifikasi serta diibaratkan sebagai entitas. Sedangkan, OOD menempatkan objek pada implementasi sebagai pemenuhan kebutuhan sistem.

Dibawah ini merupakan keuntungan dan kelemahan dari pendekatan Objek Orientasi (Nasita, 2016):

- a. Fungsi dan data dibungkus pada suatu objek sehingga lebih singkat namun berisi.

- b. Penyelesaian masalah kompleks akan lebih efektif, karena pendekatan ini memisahkan setiap kode program menjadi objek-objek, sesuai dengan fungsinya.
- c. *Tracking error* akan lebih mudah karena pemrograman lebih sederhana dengan objek.
- d. Objek yang dapat dimanfaatkan berulang kali sehingga cukup tidak terlalu banyak deklarasi fungsi.

Kelemahan yang dimiliki jika seorang pengembang terbiasa dengan pendekatan ini, maka ia akan kesulitan beradaptasi dengan pendekatan non-OO.

2.2.4.3 Pemodelan Object Oriented



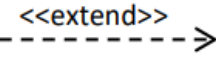

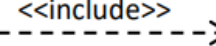
Pemodelan berorientasi objek dapat dimodelkan berupa *Unified Modeling Language* (UML) diagram. UML merupakan sebuah standar dalam merancang maupun mendokumentasikan sistem perangkat lunak. Berikut tipe-tipe UML yang berorientasi objek:


- a. *Use Case diagram*

Use Case diagram adalah diagram atau deskripsi visual dari beberapa atau semua *user* dengan interaksinya sehingga dapat menampilkan gambaran alur kerja suatu sistem. Pada diagram ini dapat diketahui seluruh fungsi yang terdapat pada sistem yang akan dibuat

Penjelasan notasi dari *Use Case Diagram* dapat dilihat pada tabel 2.1

Tabel 2.1 Notasi Use Case Diagram

Simbol	Deskripsi
	Actor adalah wujud eksternal yang nantinya berinteraksi dengan perangkat seperti pengguna atau perangkat keras.
	Association adalah hubungan yang menunjukkan interaksi antara aktor dan <i>use case</i> .
	Extend Relationship adalah hubungan antar <i>use-case</i> satu dengan yang lain. notasi ini bersifat opsional.
	Generalization Relationship adalah hubungan antar aktor satu dengan yang lain yang lebih khusus.
	Include Relationship adalah hubungan antar <i>use-case</i> satu dengan yang lain. notasi ini bersifat mandatori.

	Use Case adalah spesifikasi dari sekumpulan tindakan yang dilakukan oleh perangkat.
---	--

Sumber: (Paradigm, 2016)

b. *Use Case Scenario*

Use Case diagram yang berfungsi menggambarkan interaksi aktor dengan *use-case* secara umum namun tidak menjelaskan secara detail tentang bagaimana perangkat akan dikembangkan (Arms, 2014). Cara terbaik untuk merepresentasikan alur dari perangkat yang akan dibangun adalah dengan *Use Case Scenario*.

Penjelasan detail informasi mengenai *Use Case Scenario* dapat dilihat pada tabel 2.2.

Tabel 2.2 Detail *Use Case Scenario*

Rincian	Deskripsi
Objective	Deskripsi penggunaan <i>use-case</i>
Actor	Objek yang menerima dan mengirim informasi ke/dari <i>use-case</i>
Pre-Condition	Kondisi sebelum <i>use-case</i> diimplementasikan
Main Flow	Deskripsi langkah-langkah utama dalam sebuah perangkat
Alternative Flow	Deskripsi langkah-langkah alternatif dalam sebuah perangkat.
Post-Condition	Kondisi setelah <i>use-case</i> diimplemetasikan


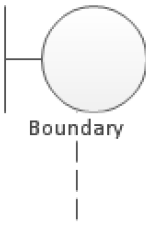



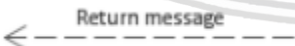

Sumber: (Miles & Hamilton, 2006)


c. *Sequence Diagram*

Sequence Diagram merupakan diagram yang mendeskripsikan secara dinamis kolaborasi objek. Bertujuan untuk memperlihatkan tahapan interaksi antar objek.

Penjelasan notasi dalam *Sequence Diagram* dapat dilihat pada tabel 2.3

Tabel 2.3 Notasi *Sequence Diagram*

Simbol	Deskripsi
	Lifeline Actor adalah wujud eksternal yang melakukan interaksi dengan perangkat
	Lifeline <<Boundary>> merupakan wujud yang akan menjembatani aktor dengan perangkat seperti monitor
	Lifeline <<Control>> merupakan wujud yang memperantarai Boundary dan Entity.
	Lifeline <<Entity>> merupakan sebuah entitas berwujud data sistem, sering dikenal sebagai Model.
	Message mendeskripsikan komunikasi antar <i>life-line</i> yang masuk/menuju perangkat.
	Return Message mendeskripsikan komunikasi antar <i>life-line</i> yang keluar/kembalian dari perangkat.
	Alternative Combined Fragment adalah sebuah ekspresi yang bersifat opsional

	Loop Combined Fragment adalah ekspresi kombinasi perulangan/iterasi
---	--

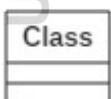


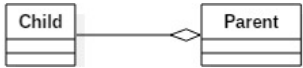

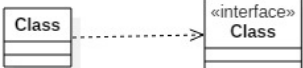
Sumber: (Paradigm, 2016)

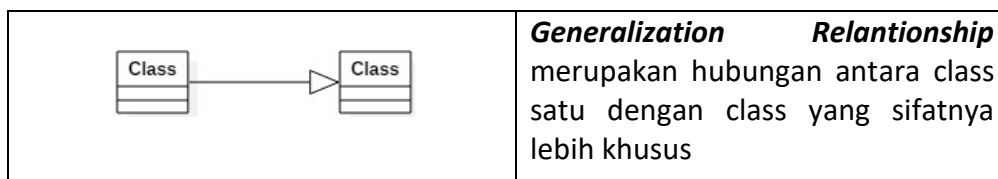
d. Class Diagram

Class Diagram merupakan deskripsi diagram mengenai struktur *class* yang terdapat di dalam sebuah perangkat (Lee, 2012). *Class* mewakili runtutan perintah yang dieksekusi oleh perangkat. Suatu *class* dapat berelasi dengan class lain dengan cara: *dependent* (suatu class tergantung atau menggunakan class yang lain), *associated* (suatu class terhubung satu sama lain), *package* (beberapa class yang digrouping menjadi satu unit), atau *specialized* (suatu class merupakan betuk khusus dari class lain), atau. Sebuah perangkat mempunyai lebih dari satu *Class Diagram*, namun tidak selalu.

Penjelasan notasi dari *Class Diagram* dapat dilihat pada tabel 2.4

Tabel 2.4 Notasi Class Diagram

Simbol	Deskripsi
	Class mendeskripsikan sekumpulan objek dan method yang berinteraksi
	Interface Class menggambarkan fitur dalam class
	Association Relationship menunjukkan relasi antar class dengan class lainnya
	Aggregation Relationship merupakan hubungan antara class yang memiliki peran sebagai parent dengan class yang memiliki peran sebagai child
	Composition Relationship mendeskripsikan bahwa class yang berperan sebagai child tidak dapat terpisah dari parent class
	Dependency Relationship sebuah class yang berhubungan dengan interface class



Sumber: Visual Paradigm (2016)

2.2.5 Pseudocode

Pseudocode adalah cara informal untuk mengekspresikan desain program komputer atau algoritma, yang bertujuan untuk mendapatkan ide dengan cepat dan juga mudah dibaca tanpa detail (Bell, 2017). *Pseudocode* mempermudah pembuatan program yang sulit dan panjang agar mudah dibaca. Sebelum adanya *pseudocode* seorang programmer menggunakan *flowchart* untuk memetakan program sebelum menulis *code*, namun kekurangan *flowchart* yaitu sulit untuk menampilkan semua bagian program dengan diagram alur. Maka dari itu *pseudocode* menutupi kekurangan tersebut dengan memetakan program sampai sedetail mungkin menggunakan bahasa apapun.

Tidak ada standart atau perintah khusus dalam menulis *pseudocode* karena *pseudocode* menerjemahkan pernyataan ke dalam bahasa apapun. Yang harus dilakukan adalah menuliskannya dengan kata-kata sendiri dalam pernyataan singkat. Berikut merupakan contoh *pseudocode* membuat program untuk menambahkan 2 angka secara bersamaan dan kemudian menampilkan hasilnya:

Tabel 2.5 Pseudocode

<p>Mulai Program Masukan dua angka, A, B Jumlahkan angka A dan B Tampilkan hasil penjumlahan Akhiri Program</p>

2.2.6 Teknologi Pengembangan Perangkat Lunak

2.2.6.1 Hyper Text Markup Language (HTML)

HTML adalah bahasa pemrograman yang mendeskripsikan struktur sebuah website (W3C, 2016). HTML merupakan protokol yang perinciannya dirumuskan oleh World Wide Web Consortium (W3C) juga komunitas WHATWG. Versi terbaru dari HTML yaitu HTML5. File HTML disimpan dengan ekstensi .html.

HTML biasanya digunakan untuk (W3C, 2016):

1. Menampilkan file yang berekstensi HTML secara online walaupun pada local komputer dengan Heading (<h1> sampai <h6>), Paragraph (<paragraph> atau <p>), Table (<table>), Lists (dan), Image (), dan sebagainya.

2. Mengambil informasi dapat dilakukan melalui tautan (`<a>`) atau mengklik tombol (`<button>`)
3. Membuat formulir memanfaatkan fitur form (`<input>`, `<textarea>`, `<select>`)

Contoh dokumen HTML sederhana:

1	<code><!DOCTYPE HTML></code>
2	<code><html></code>
3	<code><head></code>
4	<code><title>Hi, Please Welcome</title></code>
5	<code></head></code>
6	<code><body style="color: #00F; background-color: #FF0;"></code>
7	<code><h1>Hmm</h1></code>
8	<code></body></code>
9	<code></html></code>

2.2.6.2 Cascading Style Sheet (CSS)

CSS adalah bahasa pemrograman yang merepresentasikan halaman web, termasuk warna, peletakan objek, dan huruf (W3C, 2016). CSS membuat pekerjaan lebih efektif karena baik untuk penyesuaian tampilan dan *compatible* di perangkat manapun. File CSS disimpan dengan ekstensi `.css`.

Contoh dokumen CSS sederhana:

1	<code>h1 {</code>
2	<code>color: #0789de;</code>
3	<code>}</code>

2.2.6.3 Cascading Style Sheet (CSS)

JavaScript berfungsi meningkatkan fungsionalitas sebuah halaman HTML dan umumnya terdapat dalam kode HTML. Halaman web yang memiliki JavaScript lebih interaktif dan dinamis, sehingga reaksi halaman terhadap sebuah perintah memiliki respon khusus, menerima *request*, melakukan validasi data, memproduksi cookies, mengecek browser yang digunakan pengguna, dan sebagainya (Binarso, et al., 2012).

JavaScript dituliskan dalam code dengan tag `<head>` dan diawali dengan tag `<script type="text/javascript">`. Berikut contoh kode program tersebut:

1	<code><script type="text/javascript"></code>
2	<code>alert("Hmm!");</code>
3	<code></script></code>

JavaScript juga dapat disimpan di file berbeda dengan ekstensi `(.js)`. Kode tersebut dipanggil dalam tag `<script>` dengan mendeskripsikan path file pada atribut `"src"`. Berikut contoh kode program tersebut:

1	<code><script src="bootstrap.js" type="text/javascript"></script></code>
---	--

2.2.6.4 Bootstrap

Bootstrap berisi CSS, JavaScript, dan HTML untuk pemenuhan komponen web dan komponen antarmuka seperti navigasi, formulir, tombol, tabel, dan komponen lainnya. *Bootstrap* adalah salah satu front-end framework yang handal

dan siap digunakan dalam pengembangan web, sehingga lebih cepat dan efisien. *Bootstrap* juga dapat dimanfaatkan untuk membuat peletakan yang lebih responsif (Adhiasta, 2016).

2.2.6.5 Database Management System (DBMS) MySQL

MySQL adalah *Relational Database Management System* (RDBMS). RDBMS merupakan perangkat lunak yang berguna untuk mengimplementasikan basis data dalam bentuk tabel, kolom, dan indeks yang saling berhubungan antara satu tabel dengan lainnya. MySQL merupakan contoh dari RDBMS yang dikenal cepat serta mudah digunakan yang digunakan untuk perangkat lunak berskala kecil maupun besar (Binarso, et al., 2012). MySQL memiliki beberapa kelebihan dibawah ini:

- Perangkat *open-source*.
- Menggunakan bahasa standar basis data yaitu SQL.
- Compatible* pada sistem operasi dan bahasa pemrograman apapun.
- Memiliki kapasitas sampai 50 juta baris atau jika dikonversikan mencapai 8 juta TB.

Contoh sederhana penerapan MySQL:

1	SELECT * FROM Customers;
2	
3	INSERT INTO Employees (Name)
4	VALUES ('Rania');
5	
6	UPDATE Employees SET City='Depok'
7	WHERE CustomerID=11;
8	
9	DELETE FROM Employees
10	WHERE Name='Helfi Pangestu';

2.2.6.6 Hypertext Preprocessor (PHP)

PHP dikenal sebagai bahasa pemrograman server-side. Pengaplikasian PHP untuk menjalankan sebuah fungsi seperti membaca, menulis, dan menutup sebuah file, menambahkan, memodifikasi dan menghapus data dalam basis data. Juga dapat digunakan untuk mengakses dan mengontrol cookies dan session, mengontrol hak akses, dan mengenkripsi data (Binarso, et al., 2012).

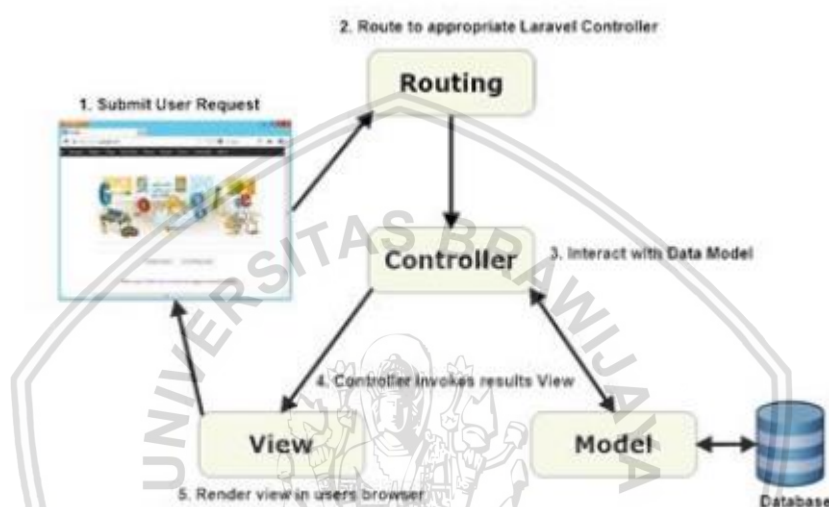
1	<?php
2	Kode;
3	?>

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain (PJ, 2017):

- Sederhana dan mudah dipahami
- Flexible* dan *compatible* pada sistem operasi apapun
- Open source
- Banyak framework yang mengadopsi bahasa ini seperti CodeIgniter dan Laravel
- Mendukung pendekatan OO

2.2.6.7 Laravel Framework

Laravel adalah sebuah framework web berbasis PHP yang open-source dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di laravel terdapat routing yang menjembatani antara request dari user dan controller. Jadi controller tidak langsung menerima request tersebut. Berikut adalah ilustrasi dari konsep MVC pada laravel yang ditunjukkan pada Gambar 2.3.



Gambar 2.2 Ilustrasi Laravel Framework

Sumber: (Yudanto, 2017)

Pada penelitian yang dilakukan Nilesh Solanki dengan judul “A Survey on different Framework of PHP” melakukan perbandingan antar *framework* PHP yaitu Code Igniter, Laravel and CakePHP mengungkapkan bahwa laravel adalah *framework* terbaik di antara *framework* PHP lainnya. Dengan dokumentasi yang mudah di baca dan memberikan ruang lingkup kreatifitas bagi pengembang dengan forum diskusi online yang aktif. Alasan utama mengapa Laravel adalah kerangka kerja terbaik dan memiliki keunggulan atas kerangka kerja lainnya adalah laravel sangat dapat dikonfigurasi, dan memiliki *engine* template blade yang cepat, pengolahan data yang cepat dan *extendable*, sehingga memudahkan pengembang Laravel untuk menambahkan fitur tanpa perlu mengganggu *core*. (Solanki, et al., 2017).

ORM Eloquent yang disertakan dengan Laravel menyediakan implementasi ActiveRecord yang baik dan sederhana untuk bekerja dengan basis data. Setiap tabel basis data memiliki "Model" yang terkait yang digunakan untuk berinteraksi dengan tabel tersebut. Model memungkinkan untuk kueri data dalam tabel, serta

memasukkan catatan baru ke dalam tabel. Adapun nama method yang sudah disediakan oleh Laravel dalam melakukan *query* terhadap model atau basis data.

Method	Contoh	Keterangan
all()	<pre><?php use App\Flight; \$flights = App\Flight::all(); foreach (\$flights as \$flight) { echo \$flight->name; }</pre>	Metode Eloquent all akan mengembalikan semua hasil dalam tabel model
get()	<pre>\$flights = App\Flight::where('active', 1) ->orderBy('name', 'desc') ->take(10) ->get();</pre>	Metode Eloquent get akan mengambil semua dan/atau mengambil beberapa hasil
find(where) atau first()	<pre>// Retrieve a model by its primary key... \$flight = App\Flight::find(1); // Retrieve the first model matching the query constraints... \$flight = App\Flight::where('active', 1)->first();</pre>	Metode Eloquent find() atau first akan mengambil data tunggal.
findOrFail() dan firstOrFail()	<pre>\$model = App\Flight::findOrFail(1); \$model = App\Flight::where('legs', '>', 100)->firstOrFail();</pre>	Metode Eloquent findOrFail(where) dan firstOrFail(where) akan mengambil hasil pertama dari kueri namun, jika tidak ada hasil yang ditemukan respons HTTP 404 secara otomatis dikirim kembali ke pengguna. Tidak perlu menulis cek eksplisit

		untuk mengembalikan 404 tanggapan saat menggunakan metode ini.
save()	<p>Contoh untuk insert:</p> <pre> <?php namespace App\Http\Controllers; use App\Flight; use Illuminate\Http\Request; use App\Http\Controllers\Controller; class FlightController extends Controller { /** * Create a new flight instance. * * @param Request \$request * @return Response */ public function store(Request \$request) { // Validate the request... \$flight = new Flight; \$flight->name = \$request- >name; \$flight->save(); } } </pre>	<p>Method Eloquent save digunakan untuk membuat <i>record</i> baru dalam database,</p> <p>Cara menggunakannya ialah membuat contoh model baru, atur atribut pada model, lalu panggil metode save()</p> <p>Metode save() juga dapat digunakan untuk memperbarui model yang sudah ada dalam database. Untuk memperbarui model, langkah pertama adalah harus mengambilnya, set semua atribut yang ingin di perbarui, dan kemudian memanggil metode save().</p>

	Contoh untuk update: \$flight = App\Flight::find(1); \$flight->name = 'New Flight Name'; \$flight->save();	
delete()	\$flight = App\Flight::find(1); \$flight->delete();	Metode Eloquent yang berfungsi untuk menghapus data

Pada penelitian ini *framework* laravel digunakan untuk membangun halaman website yang berfungsi menampilkan hasil perangkian secara *detail*.

2.2.6.8 Teknik Pengujian

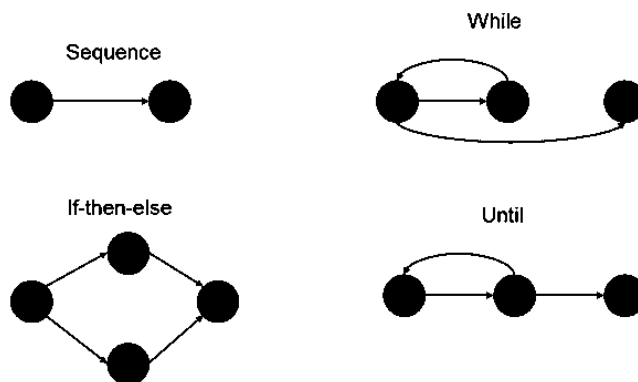
Terdapat beberapa teknik pengujian yang dapat diterapkan dalam proses pengembangan perangkat. Masing-masing teknik memiliki case testing. Dibawah ini adalah beberapa teknik yang dapat digunakan:

a. *White box Testing*

Teknik pengujian ini digunakan untuk melakukan pengujian terhadap struktur dari hasil implementasi perangkat. Salah satu jenis white-box testing, yakni *Basis Path Testing* (Pressman & Maxim, 2015). (Guru99, 2018) berpendapat, *basis path testing* adalah teknik *white box testing* yang menghasilkan kompleksitas logis (*logical complexity*) dari rancangan secara prosedural dan menggunakan pengukuran ini sebagai dasar kumpulan jalur yang akan dieksekusi. Test case yang dihasilkan dapat menjamin eksekusi setiap program paling tidak dilakukan sekali selama pengujian.

Cyclomatic Complexity adalah *software metric* yang digunakan untuk mengukur kompleksitas suatu program dengan mengukur jalur independen melalui kode program. Jalur independen adalah jalur yang minimal memiliki 1 edge yang belum pernah dilintasi sebelumnya oleh lainnya.

Notasi *flow-graph* untuk program telah ditentukan. Node dihubungkan melalui edge. Berikut adalah flow diagram sebagai pernyataan seperti *if-else*, *While*, *until*, dan *normal sequence*. Berikut gambar notasi *flow-graph* pada Gambar 2.3.



Gambar Error! No text of specified style in document..**3 Notasi Flowgraph**

Sumber:

(Guru99,

2018)

Kompleksitas suatu program dapat didefinisikan dengan:

$$V(G) = E - N + 2$$

Dimana,

E = Jumlah *edge*

N = Jumlah *node*

$$V(G) = P + 1$$

Dimana,

P = Jumlah *predicate node* (*node* yang berisi kondisi)

Cyclomatic Complexity dapat dikalkulasi secara manual pada simple program. Automated tools diperlukan jika program kompleks yang dapat melibatkan banyak *flow-graph*. Pada Tabel 2.6 dibawah ini merupakan tabel yang memberikan deskripsi kompleksitas dan arti dari nilai $V(G)$.

Tabel 2.6 Cyclomatic Complexity

Complexity Number	Meaning
1-10	Struktur dan Penulisan Baik High Testability Biaya dan Usaha Rendah
10-20	Kode Kompleks <i>Medium Testability</i> Biaya dan Usaha Medium
20-40	Kode sangat kompleks <i>Low testability</i> Biaya dan usaha tinggi
> 40	Tidak semuanya bisa diuji Biaya dan Usaha sangat tinggi

b. *Black-box Testing*

Black-box testing merupakan sebuah teknik pengujian yang ditujukan untuk menguji spesifikasi sistem tanpa diharuskan melakukan akses atau uji terhadap kode program (Sommerville, 2011).

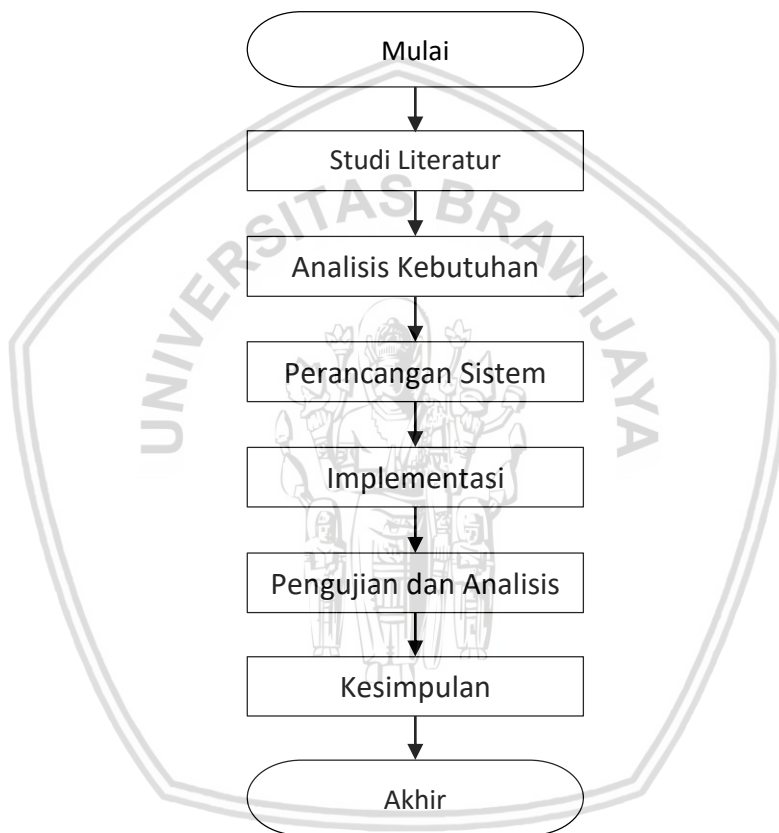
Teknik *black-box* berguna untuk mengetahui kesesuaian fungsi dari suatu perangkat dengan tujuan awal didesainnya produk tersebut. Pengujian *black-box* dilakukan untuk mendapatkan error ketika skrip dieksekusi. Menurut (Pressman, 2010, pp.495,539), *black-box testing* melakukan pencarian error dalam kategori:

1. Fungsi tidak sesuai
2. Antar muka menghasilkan kesalahan
3. *Error* saat mengakses database
4. Kesalahan behavior

Prinsip *black box testing* adalah dengan membuat sekumpulan *test case* yang telah didesain untuk mendapatkan *error* pada *level* validasi perangkat lunak (*software validation level*). Setiap *case* yang akan dijalankan, diharapkan agar dapat menemukan *error* sebanyak mungkin dengan waktu dan usaha seminimal mungkin. Setiap *test case* dapat dibuat untuk dapat melakukan uji coba terhadap setiap fungsi yang dibuat. Kemudian akan dilakukan validasi apakah telah berjalan sepenuhnya dan diwaktu yang bersamaan. Teknik pengujian dilakukan pada bagian *interface* dengan memberikan sekumpulan kondisi masukan yang akan dieksekusi secara menyeluruh sesuai kebutuhan fungsional program.

BAB 3 METODOLOGI

Bab ini berisi tahapan menyelesaikan penelitian yang berjudul “Pembangunan Aplikasi Monitoring dan Seleksi Atlet”. Penelitian ini bersifat implementatif, berupa sebuah aplikasi website untuk memonitoring atlet. Bab ini menjelaskan mengenai alur penelitian yang digunakan serta melakukan tahapan-tahapan penelitian secara spesifik dan sistematis sehingga dalam mengambil kesimpulan pada penelitian ini tidak menyimpang dari tujuan yang ditentukan sebelumnya. Berikut merupakan diagram alir yang digunakan dalam metodologi penelitian ini, dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literature merupakan tahap mempelajari teori dasar dan referensi yang mendukung sistem sebagai penunjang penelitian. Penelitian ini melakukan studi literatur terhadap pustaka dan dasar teori yang terkait dengan:

1. *Waterfall Development*

Studi literature terkait metode pengembangan daur hidup perangkat lunak dilakukan dengan mencari jurnal.

2. Spesifikasi Permainan (Basket, Voli, dan Sepak bola)

Studi literatur mengenai spesifikasi basket, voli, dan sepak bola dilakukan dengan mencari jurnal dan memahami aturan, komposisi pemain, dan teknik pada permainan.

3. Laravel Framework

Studi literatur mengenai penggunaan laravel framework dengan mencari dan mengkaji dokumentasi situs resmi.

3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan proses untuk menggambarkan sebuah sistem secara lengkap. Dalam analisis kebutuhan juga digambarkan bagaimana studi lapangan berlangsung. poin 3.2.1 menjelaskan tentang studi lapangan yang dilakukan pada penelitian ini. Analisa kebutuhan dilakukan dengan pendekatan *Object Oriented* (OO). Dibawah ini adalah penjabaran dari analisis kebutuhan yang dilakukan dengan pendekatan OO:

1. Melakukan analisis kebutuhan menjadi deskripsi dan spesifikasi kebutuhan yang nantinya akan diidentifikasi dengan aturan penomoran.
2. Hasil dari elisitasi dan analisis kebutuhan dikelompokkan menjadi tipe kebutuhan seperti fungsional dan nonfungsional
3. Mengidentifikasi aktor secara benar.
4. Menyusun pemodelan kebutuhan *Use Case Diagram* dari kebutuhan yang kebutuhan fungsional.
5. Detail *Use Case* yang terdapat pada *Use Case Diagram* dideskripsikan lebih jelas pada *Use Case Scenario*.

Hasil pemodelan *requirement* dari *Use Case Diagram* dan *Use Case Scenario* merupakan pemetaan kebutuhan fungsional yang akan digunakan dalam proses perancangan dan implementasi sistem selanjutnya.

3.2.1 Studi Lapangan

Studi lapangan dilakukan untuk mengamati kondisi yang sedang terjadi dari tim bola basket Unit Aktifitas Bola Basket Universitas Brawijaya (UABB-UB) Malang, tim sepak bola Unit Aktifitas Sepak Bola Universitas Brawijaya (UASB-UB), dan tim bola voli Unit Aktivitas Bola Volley Universitas Brawijaya (UABV-UB). Hal

tersebut dilakukan dengan mengamati kegiatan latihan yang dilakukan oleh tim tersebut. Tujuan dari kegiatan tersebut adalah untuk mengetahui faktor-faktor yang dapat menentukan pemain masuk kedalam tim utama.

Pengumpulan data pada skripsi ini menggunakan teknik sebagai berikut:

3.2.1.1 Observasi/Pengamatan

Metode observasi/pengamatan dilakukan untuk memantau secara langsung kondisi pemain dan bagaimana prosedur yang sedang diterapkan saat ini. Selain itu juga dapat mengetahui kendala yang sering terjadi terkait dengan pengambilan keputusan oleh pelatih tim bola basket, bola voli, dan sepak bola. Hal yang diamati yaitu alur pengambil keputusan dengan sumber daya pemain yang sudah ada.

3.2.1.2 Interview/Wawancara

Metode ini dilakukan dengan melakukan tanya jawab kepada pelatih dan pengurus tim bola basket, bola voli, dan sepak bola mengenai kendala yang sering dihadapi oleh tim. Hal ini dilakukan dengan tujuan untuk memperoleh informasi selengkap-lengkapny sebagai acuan dalam mengerjakan skripsi ini. Dalam wawancara ini ada beberapa pertanyaan yang diajukan mulai dari sistem pemilihan pemain yang dilakukan saat ini dan solusi yang diharapkan. Dari wawancara yang telah dilakukan didapatkan beberapa informasi dari pelatih dan juga pihak teknis UABB-UB, UASB-UB, dan UABV-UB, yaitu ada beberapa kriteria yang harus dipenuhi oleh pemain. Kriteria tersebut dibagi menjadi tiga bagian yaitu kedisiplinan, sikap, dan juga kemampuan teknis. Kriteria pertama adalah kedisiplinan yang berhubungan dengan kehadiran dalam mengikuti latihan dan juga ketepatan waktu. Kriteria selanjutnya adalah sikap. Kriteria ini perlu diperhatikan karena menjadi tolak ukur terbentuknya sebuah teamwork atau kerja sama. Sikap yang baik kepada pelatih maupun pemain akan berdampak baik pula pada kerjasama tim begitu juga sebaliknya. Dan kriteria yang terakhir adalah kemampuan teknis yang terdiri dari passing, dribbling, indurance, shoot, agility, dan footwork. Selain itu, dari wawancara ini didapatkan pula nilai bobot per kriteria yang nantinya diperlukan dalam penghitungan seleksi atlet.

3.3 Perancangan Sistem

Perancangan sistem dilakukan agar peneliti lebih mudah dalam proses implementasi. Hasil analisis kebutuhan digunakan sebagai standar dalam proses perancangan, yang dilakukan memanfaatkan pendekatan OO. Selanjutnya merupakan penjelasan perancangan sistem yang dilakukan dengan menggunakan pendekatan OO:

1. Merumuskan *Sequence Diagram* dari hasil pemodelan *Use Case Scenario*.
2. Hasil dari *Sequence Diagram* dikategorikan menjadi beberapa class pada pemodelan *Class Diagram*.

3. Merumuskan algoritma setiap fungsi pada setiap class, hasil dari pemodelan *Class Diagram* sebagai standar dalam melakukan implementasi. Sampel algoritma yang dijelaskan berjumlah 3.
4. Melakukan perancangan *Entity Relationship Diagram* secara terkonsep dan logis untuk dapat menghasilkan perancangan basis data.
5. Merancang antar muka pengguna yang akan diaplikasikan selanjutnya. Hanya ada beberapa rancangan antarmuka yang dilampirkan yang mana merupakan bagian penting dari perangkat dan akan dijadikan sampel.

Hasil perancangan *Class Diagram* beserta algoritma yang dijelaskan masing-masing fungsi dan hasil perancangan basis data ketika mengimplementasikan *Back-End* perangkat dan rancangan antarmuka akan digunakan pada saat melakukan implementasi *Front-End* perangkat.

3.4 Implementasi

Implementasi dilakukan mengacu pada analisa dan perancangan yang telah dilakukan penulis dan menyesuaikan dengan flowchart yang telah dirancang. Secara umum implementasi ini akan menghasilkan suatu website yang dibangun menggunakan *framework* laravel. Berikut ini langkah-langkah yang dirancang untuk melakukan implementasi sistem, sebagai berikut:

1. Pembuatan User Interface menggunakan html.
2. Pengimplementasian aplikasi menggunakan *codewriter*.
3. Pembuatan halaman website menggunakan laravel framework.

Sehingga dengan langkah yang dijabarkan diatas. Diharapkan implementasi ini memenuhi fungsi-fungsi dari aplikasi monitoring dan seleksi atlet UABB-UB, UABV-UB, dan UASB-UB itu sendiri yaitu menghasilkan beberapa output berupa urutan perankingan pemain berdasarkan posisi. Dari urutan tersebut akan menunjukkan pemain yang dipilih berdasarkan prioritas kriteria yang sudah diinputkan sebelumnya.

3.5 Pengujian dan Analisis

Pengujian merupakan tahap selanjutnya setelah tahap implementasi. Dalam tahap ini melakukan uji sistem dari hasil implementasi perangkat yang telah diaplikasikan. Dibawah ini merupakan penjabaran pengujian yang akan diterapkan:

1. Pengujian Unit

Pada tahap ini, pengujian dilakukan terhadap operasi yang terdapat pada setiap class. Dalam pengujian ini, *basis path testing* diterapkan untuk proses pengujian. *Basis path testing* adalah salah satu tipe dari *white-box testing*. Hanya akan terdapat tiga operasi pengujian yang akan dijadikan sampel uji.

2. Pengujian Validasi

Pada tahap ini, penguji melakukan testing terhadap perangkat untuk mengetahui apakah telah dikembangkan sesuai dengan setiap kebutuhan yang telah ditentukan pada tahap penjabaran kebutuhan fungsional dan non fungsional. Dalam pengujian ini, memanfaatkan *Equivalence partitioning*. *Equivalence partitioning* merupakan jenis dari *black-box testing*.

3. Pengujian *Compatibility*

Pada tahap ini, pengujian digunakan untuk mengetahui apakah sistem yang telah dikembangkan dapat berjalan pada hardware, sistem operasi, aplikasi, ataupun lingkungan jaringan yang berbeda. Dalam pengujian ini, memanfaatkan tiga jenis browser yaitu *Opera Browser*, *Microsoft Edge Browser*, dan *Chrome Browser*.

3.6 Kesimpulan

Kesimpulan dilakukan setelah melalui keseluruhan tahapan identifikasi masalah, studi pustaka, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi, pengujian dan juga analisis hasil. Kesimpulan merupakan hasil dari pengujian dan analisis metode untuk menjawab rumusan masalah dapat terselesaikan atau tidak. Tahap akhir penulisan penelitian ini adalah penyampaian saran yang bertujuan memberikan pertimbangan dan pandangan bagi penelitian selanjutnya untuk memperbaiki kesalahan yang ada.

BAB 4 ANALISIS KEBUTUHAN

Rekayasa kebutuhan merupakan tahapan pertama yang dilakukan pada pengembangan sebuah sistem. Tahap ini merupakan sebuah tahapan yang dilakukan untuk menentukan kebutuhan apa saja yang harus terdapat pada Pembangunan Aplikasi Monitoring dan Seleksi Atlet. Pada tahap ini akan menjelaskan gambaran umum sistem, kebutuhan-kebutuhan apa saja yang dibutuhkan oleh sistem dan siapa saja yang berinteraksi dengan sistem.

1.1 Elisitasi Kebutuhan

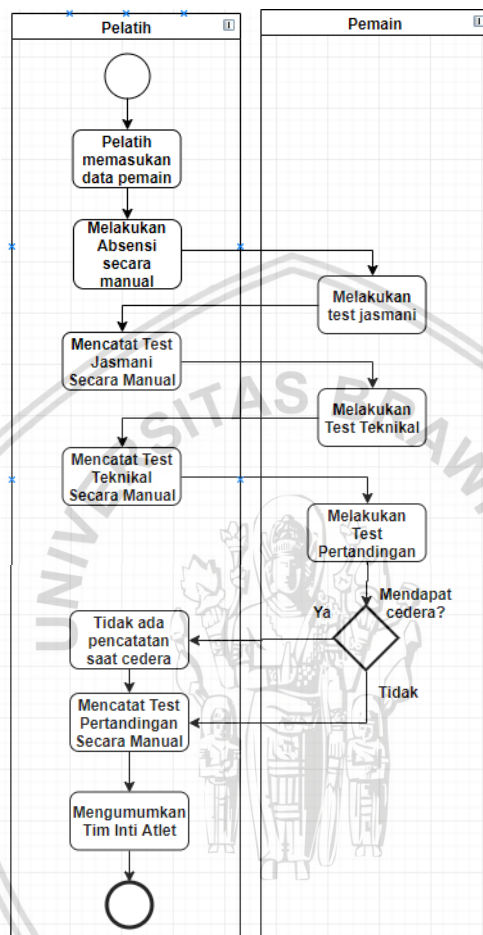
Tahap ini dilakukan untuk mengetahui masalah yang terjadi serta mengenali pemangku kepentingan dalam sistem. Tahapan ini dilakukan dengan wawancara secara langsung, yaitu wawancara dengan pelatih basket UASB, pelatih bola voli UABV, dan pelatih bola UASB untuk mengetahui permasalahan lebih mendalam. Berdasarkan hasil wawancara, maka ditemukan beberapa kendala yang dihadapi pelatih dalam mengevaluasi pemain. Kendala pertama, tidak adanya sistem pencatatan terorganisir yang terjadi saat ini masih menggunakan kertas manual sehingga memungkinkan hilang dan/atau rusaknya *document* tersebut. Kendala kedua, tidak adanya pencatatan peningkatan kemampuan atlet yang terorganisir sehingga memungkinkan terjadinya pemain yang stagnan atau tidak berkembang. Kendala ketiga, tidak adanya pencatatan riwayat Riwayat Cedera Atlet sehingga memungkinkan terjadinya cedera fatal yang diakibatkan beban latihan terlalu berat yang diberikan kepada atlet. Kendala keempat, tidak adanya penilaian kemampuan atlet secara spesifik dan tertulis sehingga memungkinkan pelatih tidak dapat mengambil keputusan yang baik dalam menentukan pemain tim inti.

Berdasarkan kendala tersebut, sistem menyediakan beberapa solusi yang diharapkan dapat menangani permasalahan yang terjadi. Solusi yang disediakan yaitu sistem aplikasi berbasis website, fitur monitoring, fitur *injury*, dan fitur *top* atlet. Yang pertama, sistem aplikasi dibangun menggunakan website dimana terdapat sebuah *dashboard* pelatih untuk menyimpan data-data atlet serta biodata pelatih itu sendiri. Website yang dibangun secara *responsive* terhadap layar *handphone* agar dapat memudahkan pelatih dalam mengoperasikan sistem dimana saja dan kapan saja.

Untuk fitur monitoring, pelatih dapat membuat suatu aktifitas apabila akan melakukan permainan seperti *excercise*, *sparing*, dan *tournament*. Sistem akan menyediakan *form* penilaian berdasarkan parameter dan *role* atlet tersebut sehingga antar pemain mendapatkan bobot yang sesuai dengan *role*-nya. Selanjutnya aktifitas yang disimpan akan tersimpan di basis data yang kemudian akan digunakan untuk mengkalkulasi peningkatan masing-masing atlet.

Untuk fitur *injury*, sistem akan menampilkan detail semua riwayat Riwayat Cedera Atlet sehingga pelatih dapat memberikan bobot yang tepat kepada masing-masing pemain. Fitur *injury* juga dilengkapi dengan gambar dan waktu cedera agar memberikan informasi yang tepat.

Untuk fitur *top* atlet, sistem akan menampilkan hasil kalkulasi setiap aktifitas yang sudah dilakukan dalam bentuk tabel. Sistem akan mengurutkan tabel *top* atlet dari atlet yang memiliki score rata-rata terbesar sampai rata-rata terkecil. Sistem juga mampu memfilter berdasarkan *role* masing-masing atlet sehingga dapat terpilih atlet yang berkualitas. Pada Gambar 4.1 merupakan proses bisnis seleksi tim inti yang terjadi saat ini.



Gambar 0.1 Proses Bisnis Seleksi Tim Inti

Pada gambar 4.1 terlihat jelas bahwa proses bisnis seleksi tim inti masih dilakukan secara manual dan belum adanya pencatatan yang terorganisir dengan baik serta tidak adanya pencatatan ketika terjadi cedera terhadap atlet.

4.1.1 Identifikasi Aktor

Identifikasi aktor bertujuan untuk mengidentifikasi aktor-aktor yang berinteraksi dengan perangkat. Pada pengembangan Pembangunan Aplikasi Monitoring dan Seleksi Atlet ini terdapat 3 Aktor, yaitu pengguna, pelatih, admin. Di dalam tabel ini disebutkan aktor-aktor yang melakukan interaksi sistem dan kolom deskripsi untuk menjelaskan peran dari setiap aktor. Proses identifikasi aktor dapat dilihat pada tabel 4.1

Tabel 0.1 Table Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Aktor yang hanya dapat melihat halaman awal. Serta melakukan login, register, dan reset password.
Pelatih	Aktor yang dapat masuk dan keluar dari sistem, mengganti kata sandi dan profil pengguna, membuat tim dan atlet, melakukan kegiatan monitoring, melakukan edit profile tim dan menambah serta mengedit penghargaan tim, melakukan edit profile atlet, menambah dan mengedit penghargaan atlet, menambah dan mengedit riwayat Cedera Atlet.
Admin	Aktor yang dapat melakukan hal yang dilakukan oleh Pengguna dan Pelatih. Serta melakukan penambahan, edit, dan hapus cabang olahraga, melakukan penambahan, edit, dan hapus cabang parameter, melakukan penambahan.

4.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan deskripsi fitur yang harus disediakan oleh perangkat serta bagaimana sebuah sistem harus memberi reaksi terhadap suatu *request* yang masuk, dan sistem harus dapat pada *multiple* kondisi. Definisi dan deskripsi kebutuhan fungsional yang dihasilkan dapat dilihat pada Tabel 4.2. Setiap definisi dan deskripsi kebutuhan akan diberikan kode "MA_AA_DD". "AA" merupakan nomor definisi kebutuhan, sedangkan "DD" merupakan nomor spesifikasi kebutuhan.

Tabel 0.2 Definisi dan Spesifikasi Kebutuhan Fungsional Sistem

NO	Use Case	Aktor	Definisi atau Spesifikasi	Kode
1	<i>Login</i>	Pengguna, Pelatih, dan Admin	Sistem harus dapat mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan cara memasukan Nama Pengguna dan Kata Sandi	MA_01_00
2	<i>logout</i>	Pelatih, dan Admin	Sistem harus dapat mengizinkan pelatih dan admin untuk keluar dari sistem	MA_02_00
3	<i>register</i>	Pengguna, Pelatih,	Sistem harus dapat mengizinkan pengguna	MA_03_00

		dan Admin	untuk mendaftar agar dapat masuk kedalam sistem.	
4	<i>reset password</i>	Pengguna, Pelatih, dan Admin	Sistem harus dapat mengizinkan pengguna yang lupa password dapat masuk kembali kedalam sistem.	MA_04_00
5	Melihat daftar top tim	Pengguna, Pelatih, dan Admin	Sistem harus dapat menampilkan daftar <i>top</i> tim pada pengguna	MA_05_00
5.1	Melihat daftar top tim	Pengguna, Pelatih, dan Admin	Data daftar <i>top</i> tim yang ditampilkan hanya foto tim, nama tim, dan score rata-rata tim selama 3 bulan terakhir.	MA_05_01
6	Melihat daftar top atlet	Pengguna, Pelatih, dan Admin	Sistem harus dapat menampilkan daftar top atlet	MA_06_00
6.1	Melihat daftar top atlet	Pengguna, Pelatih, dan Admin	Data daftar <i>top</i> atlet yang ditampilkan hanya nama atlet, posisi, dan score rata-rata pemain selama 3 bulan terakhir.	MA_06_01
7	Melihat profile Pelatih	Pelatih dan Admin	Sistem harus dapat menampilkan profile pelatih	MA_07_00
8	Melihat daftar <i>achievement</i> pelatih	Pelatih dan Admin	Data melihat daftar <i>achievement</i> pelatih yang ditampilkan yaitu foto, nama, tingkat <i>achievement</i> , dan deskripsi	MA_08_00
9	Menambahkan <i>achievement</i> pelatih	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan <i>achievement</i> pelatih	MA_09_00

10	Melihat detail <i>achievement</i> pelatih	Pelatih dan Admin	Sistem harus dapat menampilkan detail <i>achievement</i> pelatih	MA_10_00
11	Mengubah <i>achievement</i> pelatih	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah <i>achievement</i> pelatih	MA_11_00
12	Menghapus <i>achievement</i> pelatih	Pelatih dan Admin	Sistem harus menyediakan fitur menghapus <i>achievement</i> pelatih	MA_12_00
13	Mengubah profile pelatih	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk Mengubah profile pelatih.	MA_13_00
13.1	Mengubah profile pelatih	Pelatih dan Admin	Data mengubah profile pelatih yang ditampilkan yaitu foto, nama, username, email, password, jenis-kelamin, tanggal lahir, alamat, dan nomor telepon	MA_13_01
14	Melihat detail monitoring	Pelatih dan Admin	Sistem harus dapat menampilkan tim yang di monitor	MA_14_00
15	Menambah monitoring	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambah tim yang di monitor	MA_15_00
16	Melihat daftar atlet hasil monitoring aktifitas <i>exercise</i>	Pelatih dan Admin	Sistem harus dapat menampilkan daftar atlet dari hasil monitor aktifitas <i>exercise</i>	MA_16_00
17	Melihat daftar atlet hasil monitoring aktifitas <i>sparing</i>	Pelatih dan Admin	Sistem harus dapat menampilkan daftar atlet dari hasil monitor aktifitas <i>sparing</i>	MA_17_00

18	Melihat daftar atlet hasil monitoring aktifitas <i>championship</i>	Pelatih dan Admin	Sistem harus dapat menampilkan daftar atlet dari hasil monitor aktifitas <i>championship</i>	MA_18_00
19	Melihat daftar rekomendasi atlet	Pelatih dan Admin	Sistem harus dapat menampilkan daftar rekomendasi atlet.	MA_19_00
19.1	Melihat daftar rekomendasi atlet	Pelatih dan Admin	Data rekomendasi atlet yang ditampilkan yaitu score beserta parameter <i>activity</i> atlet selama 3 bulan terakhir	MA_19_01
20	Melihat daftar tim	Pelatih dan Admin	Sistem harus dapat menampilkan daftar tim.	MA_20_00
20.1	Melihat daftar tim.	Pelatih dan Admin	Data melihat daftar tim yang ditampilkan yaitu foto, nama, deskripsi, alamat, kota, dan provinsi.	MA_20_01
21	Melihat detail tim.	Pelatih dan Admin	Sistem harus menampilkan detail tim.	MA_21_00
22	Menambahkan tim.	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan tim.	MA_22_00
23	Mengubah tim.	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah tim.	MA_23_00
24	Menghapus tim.	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menghapus tim.	MA_24_00
25	Melihat daftar <i>achievement</i> tim	Pelatih dan Admin	Sistem harus dapat menampilkan daftar <i>achievement</i> tim	MA_25_00



25.1	Melihat daftar <i>achievement</i> tim	Pelatih dan Admin	Data melihat daftar <i>achievement</i> tim yang ditampilkan yaitu foto, nama, tingkat <i>achievement</i> , dan deskripsi	MA_25_01
26	Melihat detail <i>achievement</i> tim	Pelatih dan Admin	Sistem harus dapat menampilkan detail <i>achievement</i> tim	MA_26_00
27	Menambahkan <i>achievement</i> tim	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan <i>achievement</i> tim	MA_27_00
28	Mengubah <i>achievement</i> tim	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah <i>achievement</i> tim	MA_28_00
29	Menghapus <i>achievement</i> tim	Pelatih dan Admin	Sistem harus menyediakan fitur menghapus <i>achievement</i> tim	MA_29_00
30	Melihat daftar atlet	Pelatih dan Admin	Sistem harus dapat menampilkan daftar atlet	MA_30_00
30.1	Melihat daftar atlet	Pelatih dan Admin	Data daftar atlet yang di tampilkan yaitu foto, nama, jenis kelamin, nomor telepon, dan nomor punggung	MA_30_01
31	Melihat detail atlet	Pelatih dan Admin	Sistem harus dapat menampilkan detail atlet	MA_31_00
32	Menambahkan atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan atlet	MA_32_00
33	Mengubah atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah atlet	MA_33_00



34	Menghapus atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menghapus atlet	MA_34_00
35	Melihat daftar <i>achievement</i> atlet	Pelatih dan Admin	Sistem harus dapat menampilkan daftar <i>achievement</i> atlet	MA_35_00
35.1	Melihat daftar <i>achievement</i> atlet	Pelatih dan Admin	Data melihat daftar <i>achievement</i> atlet yang ditampilkan yaitu foto, nama, tingkat <i>achievement</i> , dan deskripsi	MA_35_01
36	Menambahkan <i>achievement</i> atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan <i>achievement</i> atlet	MA_36_00
37	Melihat detail <i>achievement</i> atlet	Pelatih dan Admin	Sistem harus dapat menampilkan detail <i>achievement</i> atlet	MA_37_00
38	Mengubah <i>achievement</i> atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah <i>achievement</i> atlet	MA_38_00
39	Menghapus <i>achievement</i> atlet	Pelatih dan Admin	Sistem harus menyediakan fitur menghapus <i>achievement</i> atlet	MA_39_00
40	Melihat daftar riwayat cedera atlet	Pelatih dan Admin	Sistem harus dapat menampilkan daftar riwayat cedera atlet	MA_40_00
41	Melihat detail riwayat cedera atlet	Pelatih dan Admin	Sistem harus dapat menampilkan detail riwayat cedera atlet	MA_41_00
42	Menambahkan riwayat cedera atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan riwayat cedera atlet	MA_42_00

43	Mengubah riwayat cedera atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah riwayat cedera atlet	MA_43_00
44	Menghapus riwayat cedera atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menghapus riwayat cedera atlet	MA_44_00
45	Melihat Daftar Aktifitas Atlet	Pelatih dan Admin	Sistem harus dapat menampilkan daftar aktifitas atlet	MA_45_00
45.1	Melihat Daftar Aktifitas Atlet	Pelatih dan Admin	Data melihat daftar aktifitas atlet yang ditampilkan yaitu waktu, tempat, dan tipe pertandingan.	MA_45_01
46	Melihat detail parameter penilaian	Pelatih dan Admin	Sistem harus dapat menampilkan detail parameter penilaian	MA_46_00
46.1	Melihat detail parameter penilaian	Pelatih dan Admin	Data melihat detail parameter penilaian yang ditampilkan yaitu daftar parameter tetap penilaian atlet, daftar parameter tidak tetap penilaian atlet, dan persentase tipe parameter penilaian.	MA_46_01
47	Melihat daftar parameter tetap penilaian atlet	Pelatih dan Admin	Sistem harus dapat menampilkan parameter tetap penilaian atlet	MA_47_00
48	Melihat daftar parameter tidak tetap penilaian atlet	Pelatih dan Admin	Sistem harus dapat menampilkan parameter tidak tetap penilaian atlet	MA_48_00
49	Menambahkan parameter tidak tetap penilaian atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan	MA_49_00

			parameter tidak tetap penilaian atlet	
50	Mengubah parameter tidak tetap penilaian atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah parameter tidak tetap penilaian atlet	MA_50_00
51	Menghapus parameter tidak tetap penilaian atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk menghapus parameter tidak tetap penilaian atlet	MA_51_00
52	Mengubah persentase tipe parameter penilaian	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah persentase tipe parameter penilaian	MA_52_00
53	Melihat daftar pelatih	Admin	Sistem harus dapat menampilkan daftar pelatih	MA_53_00
54	Melihat detail pelatih	Admin	Sistem harus dapat menampilkan detail pelatih	MA_54_00
55	Menambahkan pelatih	Admin	Sistem harus dapat mengizinkan admin untuk menambahkan pelatih	MA_55_00
56	Mengubah pelatih	Admin	Sistem harus dapat mengizinkan admin untuk mengubah pelatih	MA_56_00
57	Menghapus pelatih	Admin	Sistem harus dapat mengizinkan admin untuk menghapus pelatih	MA_57_00
58	Melihat daftar cabang olahraga	Admin	Sistem harus dapat menampilkan daftar cabang olahraga	MA_58_00
59	Melihat detail cabang olahraga	Admin	Sistem harus dapat menampilkan detail cabang olahraga	MA_59_00

60	Menambahkan cabang olahraga	Admin	Sistem harus dapat mengizinkan pengguna untuk menambahkan cabang olahraga	MA_60_00
61	Mengubah cabang olahraga	Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah cabang olahraga	MA_61_00
62	Menghapus cabang olahraga	Admin	Sistem harus dapat mengizinkan pengguna untuk menghapus cabang olahraga	MA_62_00
63	Melihat Tampilan <i>Home</i>	Pengguna, Pelatih, dan Admin	Sistem harus dapat menampilkan tampilan <i>Home</i> yang berisi informasi dan hal-hal terkait sistem athmonitor	MA_63_00
64	Melihat Tampilan <i>Dashboard</i>	Pelatih dan Admin	Sistem harus dapat menampilkan tampilan <i>Dashboard</i>	MA_64_00
65	Mengubah <i>score</i> atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengubah <i>score</i> atlet	MA_65_00
66	Mengakhiri monitoring atlet	Pelatih dan Admin	Sistem harus dapat mengizinkan pengguna untuk mengakhiri monitoring atlet	MA_66_00

4.3 Kebutuhan Non Fungsional

Kebutuhan Non fungsional merupakan faktor pendukung yang dapat menjadikan aplikasi atau sistem yang dibuat dapat berjalan sesuai dengan yang diinginkan. Tabel 4.3 menunjukan kebutuhan non fungsional sistem yang harus dipenuhi agar sistem dapat berjalan dengan baik

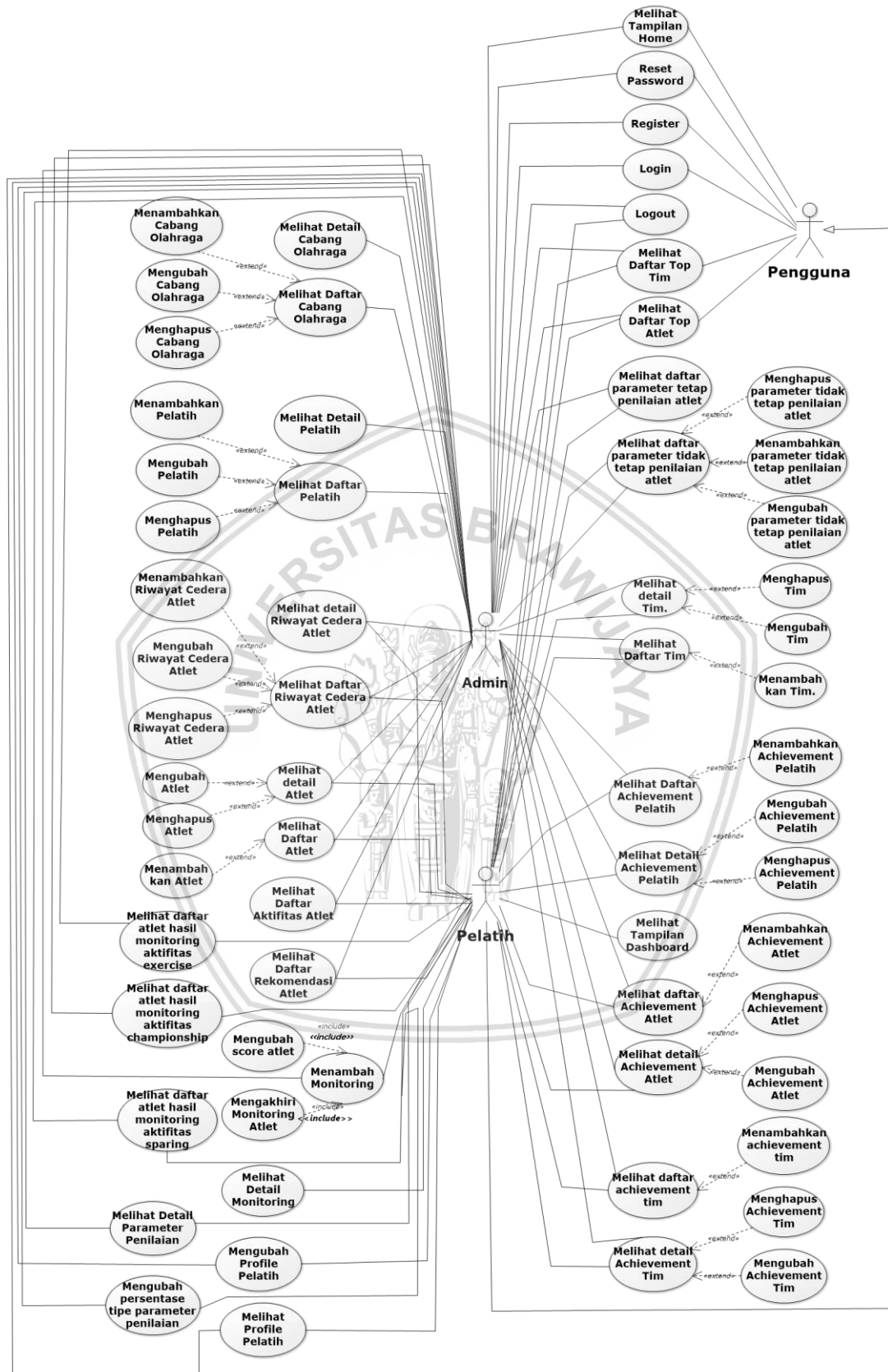
Tabel 0.3 Daftar Kebutuhan Non Fungsional

NO	Parameter	Deskripsi Kebutuhan
1	Compatibility	Sistem dapat berjalan dengan baik pada Opera Browser, Chrome Browser, dan Microsft Edge Browser

4.4 Use Case Diagram

Setelah melakukan elisitasi kebutuhan, selanjutnya membuat pemodelan *Use Case Diagram*. Terdapat 3 aktor dan 66 kebutuhan fungsional yang dihasilkan pada tahap elisitasi kebutuhan. Hasil dari elisitasi kebutuhan tersebut dijadikan dasar dalam membuat *Use Case Diagram*. *Use Case Diagram* dapat dilihat pada Gambar 4.2





Gambar 0.2 Use Case Diagram Sistem

4.5 Use Case Scenario

Setelah *Use Case diagram* telah dibuat, selanjutnya menjelaskan masing-masing *Use Case* yang terdapat pada *Use Case diagram* dengan membuat *Use Case Scenario*. Berikut *Use Case Scenario* dari masing-masing *Use Case* tersebut:

4.5.1 Login

Use case scenario login dapat dilihat pada Tabel 4.3

Tabel 0.4 Use Case Scenario Login

Action	Pengguna, Pelatih, dan Admin
Objective	Fungsi untuk dapat masuk ke dalam sistem
Pre-Condition	Halaman login berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> dan <i>password</i> dan memilih tombol <i>login</i> 2. Sistem mengecek status validitas input dari pengguna kemudian menampilkan halaman <i>dashboard</i>
Alternative Flow: Jika <i>username</i> dan/atau <i>password</i> salah	1. Jika Nama Pengguna dan/atau Kata Sandi salah, maka sistem akan menampilkan pesan "Provided username and password does not match!"
Post-Condition	Sistem menampilkan halaman menu

4.5.2 Logout

Use case scenario logout dapat dilihat pada Tabel 4.4

Tabel 0.5 Use Case Scenario Logout

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat keluar ke dalam sistem
Pre-Condition	Aktor telah masuk ke dalam sistem sebagai pemilik atau pengguna dan berada pada halaman <i>dashboard</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>logout</i> 2. Sistem menghapus session pengguna dan menampilkan halaman <i>login</i>
Alternative Flow	-
Post-Condition	Sistem menghapus session pengguna dan menampilkan halaman <i>login</i>

4.5.3 Register

Use case scenario register dapat dilihat pada Tabel 4.5

Tabel 0.6 Use Case Scenario Register

Action	Pengguna, pelatih dan Admin
Objective	Fungsi untuk dapat mendaftarkan diri ke dalam sistem
Pre-Condition	Halaman register berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i>, password, email, sport, gender, dan mencentang term of policy. Aktor menekan tombol Register. 2. Sistem mengecek <i>username</i> dan/atau email digunakan sebelumnya atau belum, jika belum maka sistem menyimpan masukkan actor dan menampilkan pesan bahwa registrasi berhasil.
Alternative Flow : Jika username atau email telah digunakan sebelumnya	1. Jika <i>username</i> atau email sudah terpakai maka sistem akan menampilkan pesan error.
Post-Condition	Sistem menyimpan masukkan actor, menampilkan pesan "Register successfully!" dan menampilkan halaman login

4.5.4 Reset Password

Use case scenario Reset Password dapat dilihat pada Tabel 4.6

Tabel 0.7 Use Case Scenario Reset Password

Action	Pengguna, Pelatih, dan Admin
Objective	Fungsi untuk dapat melakukan reset ulang Kata Sandi agar dapat masuk ke dalam sistem
Pre-Condition	Halaman reset password berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memasukkan email dan menekan tombol <i>reset password</i>. 2. Sistem akan mengecek email tersebut sudah terpakai atau belum, jika sudah maka sistem akan mengirimkan halaman reset password melalui email dan menampilkan pesan bahwa reset password berhasil.
Alternative Flow: Jika email belum terdaftar kedalam sistem	Sistem akan menampilkan pesan "Your email address was not found."

Post-Condition	Sistem akan mengirimkan link untuk mengubah kata sandi kedalam email.
-----------------------	---

4.5.5 Melihat Daftar *Top Tim*

Use case scenario melihat daftar top tim dapat dilihat pada Tabel 4.7

Tabel 0.8 Use Case Scenario Melihat Daftar Top Tim

Action	Pengguna, Pelatih, Admin
Objective	Fungsi untuk dapat melihat 3 tim dengan urutan nilai terbaik berdasarkan cabang olahraga.
Pre-Condition	Halaman awal berhasil ditampilkan.
Main Flow	1. Aktor menekan tombol <i>top tim</i> pada halaman <i>home</i> . 2. Sistem akan menampilkan halaman <i>top tim</i> .
Alternative Flow	-
Post-Condition	Sistem akan menampilkan informasi halaman <i>top team</i> .

4.5.6 Melihat Daftar *Top Atlet*

Use case scenario melihat daftar top atlet dapat dilihat pada Tabel 4.8

Tabel 0.9 Use Case Scenario Melihat Daftar Top Atlet

Action	Pengguna, Pelatih, Admin
Objective	Fungsi untuk dapat melihat 3 atlet dengan urutan nilai terbaik berdasarkan cabang olahraga.
Pre-Condition	Halaman awal berhasil ditampilkan.
Main Flow	1. Aktor menekan tombol daftar <i>top atlet</i> pada halaman <i>home</i> . 2. Sistem akan menampilkan halaman daftar <i>top atlet</i> .
Alternative Flow	-
Post-Condition	Sistem akan menampilkan informasi halaman <i>top atlet</i> .

4.5.7 Melihat Tampilan *Home*

Use case scenario melihat tampilan *home* dapat dilihat pada tabel 4.9

Tabel 0.10 Use Case Scenario Melihat Tampilan Home

Action	Pengguna, Pelatih, Admin
Objective	Fungsi untuk dapat melihat halaman awal dari website.
Pre-Condition	Halaman awal berhasil ditampilkan
Main Flow	1. Aktor memasukan nama website. 2. Sistem akan menampilkan halaman awal.

Alternative Flow	-
Post-Condition	Sistem akan menampilkan halaman awal website

4.5.8 Melihat Tampilan *Dashboard*

Use case scenario melihat tampilan *dashboard* dapat dilihat pada tabel 4.10

Tabel 0.11 Melihat Tampilan *Dashboard*

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat melihat halaman <i>dashboard</i>
Pre-Condition	Aktor sudah memiliki otoritas sebagai pelatih atau admin
Main Flow	1. Aktor menekan tombol <i>Home</i> saat sudah masuk ke dalam sistem 2. Sistem akan menampilkan halaman <i>dashboard</i>
Alternative Flow	-
Post-Condition	Sistem akan menampilkan informasi halaman awal website

4.5.9 Melihat Profile Pelatih

Use case scenario Melihat Profile Pelatih dapat dilihat pada tabel 4.11

Tabel 0.12 Use Case Scenario Melihat Profile Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat Melihat Profile Pelatih
Pre-Condition	Halaman <i>Dashboard</i> berhasil di tampilkan
Main Flow	1. Aktor menekan tombol Profile. 2. Sistem menampilkan halaman profile pelatih
Alternative Flow	-
Post-Condition	Profile pelatih berhasil ditampilkan

4.5.10 Mengubah Profile Pelatih

Use case scenario mengubah profile pelatih dapat dilihat pada tabel 4.12

Tabel 0.13 Use Case Scenario Mengubah Profile Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah profile pelatih
Pre-Condition	Halaman profile pelatih berhasil di tampilkan
Main Flow	1. Aktor menekan tombol edit profiles. 2. Sistem menampilkan formulir mengubah profile pelatih.

	<ol style="list-style-type: none"> Aktor mengubah Username, Email, Password, Gender, Birth of Date, Address, Phone, dan Avatar dan menekan tombol submit. Sistem menyimpan data profile pelatih ke dalam basis data dan menampilkan pesan "Update user successfully!"
Alternative Flow: Jika username telah digunakan atau email telah digunakan atau phone telah digunakan	Sistem akan menampilkan pesan error dan menampilkan halaman profile pelatih.
Post-Condition	Sistem berhasil diperbarui dan menampilkan profile pelatih

4.5.11 Melihat Daftar *Achievement* Pelatih

Use case scenario melihat daftar *achievement* pelatih dapat dilihat pada tabel 4.13

Tabel 0.14 Use Case Scenario Melihat Daftar Achievement Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat melihat daftar <i>achievement</i> pelatih.
Pre-Condition	Halaman Melihat Profile Pelatih berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> Aktor menekan <i>tab achievement</i> Sistem akan menampilkan daftar <i>achievement</i> pelatih
Alternative Flow	-
Post-Condition	Daftar <i>achievement</i> pelatih berhasil ditampilkan

4.5.12 Menambahkan *Achievement* Pelatih

Use case scenario menambahkan *achievement* pelatih dapat dilihat pada tabel 4.14

Tabel 0.15 Use Case Scenario Menambah Achievement Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menambahkan <i>achievement</i> pelatih
Pre-Condition	Halaman profile pelatih berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> Aktor menekan tombol <i>add achievement</i>. Sistem menampilkan formulir tambah <i>achievement</i> pelatih. Aktor mengisi form tambah <i>achievement</i> berupa name, description, image, level, date, coach dan menekan tombol submit.

	4. Sistem menyimpan data <i>achievement</i> pelatih ke dalam basis data dan menampilkan pesan “Add achievement successfully!”
Alternative Flow:	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> pelatih

4.5.13 Melihat Detail *Achievement* Pelatih

Use case scenario melihat detail *achievement* pelatih dapat dilihat pada tabel 4.15

Tabel 0.16 Melihat Detail *Achievement* Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail <i>achievement</i> pelatih.
Pre-Condition	Halaman profile pelatih berhasil di tampilkan
Main Flow	1. Aktor menekan tombol lihat pada tab <i>achievement</i> 2. Sistem akan menampilkan halaman detail <i>achievement</i> pelatih
Alternative Flow	-
Post-Condition	Halaman detail <i>achievement</i> pelatih berhasil ditampilkan

4.5.14 Mengubah *Achievement* Pelatih

Use case scenario mengubah *achievement* pelatih dapat dilihat pada tabel 4.16

Tabel 0.17 Use Case Scenario Mengubah *Achievement* Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah <i>achievement</i> pelatih
Pre-Condition	Halaman detail <i>achievement</i> pelatih berhasil ditampilkan
Main Flow	1. Aktor menekan tombol edit <i>achievement</i> . 2. Sistem menampilkan formulir mengubah <i>achievement</i> pelatih. 3. Aktor mengisi form mengubah <i>achievement</i> pelatih berupa name, description, image, level, date, coach dan menekan tombol submit. 4. Sistem menyimpan data <i>achievement</i> pelatih ke dalam basis data dan menampilkan pesan “Update achievement successfully!”
Alternative Flow:	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> pelatih

4.5.15 Menghapus *Achievement* Pelatih

Use case scenario menghapus *achievement* pelatih dapat dilihat pada tabel 4.17

Tabel 0.18 Use Case Scenario Menghapus Achievement Pelatih

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menghapus <i>achievement</i> pelatih
Pre-Condition	Halaman detail <i>achievement</i> pelatih berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>delete achievement</i>. 2. Sistem menampilkan konfirmasi menghapus <i>achievement</i> pelatih. 3. Aktor menekan tombol "Yes". 4. Sistem menghapus data <i>achievement</i> atlet dari basis data, menampilkan pesan "Delete achievement successfully!", dan menampilkan halaman <i>dashboard</i>
Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> pelatih dan menampilkan halaman detail <i>achievement</i> pelatih
Post-Condition	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>

4.5.16 Melihat Detail Monitoring

Use case scenario melihat detail monitoring dapat dilihat pada tabel 4.18

Tabel 0.19 Use Case Scenario Melihat Detail Monitoring

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail monitoring.
Pre-Condition	Aktor sudah menambah aktifitas monitoring
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu monitor pada menu 2. Sistem menampilkan data daftar atlet yang akan di monitor, tanggal, tempat, tipe, dan status aktifitas monitoring
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman detail monitoring

4.5.17 Menambah Monitoring

Use case scenario menambah monitoring dapat dilihat pada tabel 4.19

Tabel 0.20 Use Case Scenario Menambah Monitoring

Action	Pelatih dan Admin
---------------	-------------------

Objective	Fungsi untuk dapat menambah monitoring
Pre-Condition	Halaman monitoring berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol add activity. 2. Sistem menampilkan formulir tambah monitoring, aktor mengisi date, place, memilih type, memilih status dan menekan tombol submit. 3. Sistem menyimpan data monitoring ke dalam basis data, sistem mengambil data atlet dari database, menampilkan pesan "Add activity successfully!", dan menampilkan halaman detail monitoring
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan detail monitoring.

4.5.18 Mengubah Score Atlet

Use case scenario mengubah score atlet dapat dilihat pada Tabel 4.20

Tabel 0.21 Use Case Scenario Mengubah Score Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk mengubah score atlet.
Pre-Condition	Halaman detail monitoring berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih atlet yang akan di monitor, menekan tombol edit, dan akan menampilkan halaman scoring 2. Aktor mengisi data scoring berupa parameter bertipe skill, attitude, dan additional. Kemudian menekan tombol submit. 3. Sistem akan menyimpan masukkan aktor dan menampilkan pesan "Update score successfully!" kemudian sistem akan menampilkan halaman detail monitoring
Alternative Flow: Jika aktor menekan tombol cancel pada halaman scoring	Sistem akan mengagalkan masukkan pengguna dan akan menampilkan halaman scoring
Alternative Flow: Jika Tidak terdapat error pada saat memproses	Sistem akan menampilkan pesan error dan menampilkan halaman scoring

method input di CA_score	
Post-Condition	Sistem berhasil diperbarui, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.

4.5.19 Mengakhiri Monitoring Atlet

Use case scenario mengakhiri monitoring atlet dapat dilihat pada Tabel 4.21

Tabel 0.22 Use Case Scenario Mengakhiri Monitoring Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk mengubah score atlet.
Pre-Condition	Halaman detail monitoring berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol End Activity, menampilkan formulir konfirmasi end activity, dan menekan tombol "Yes" 2. Sistem akan menyimpan aktifitas monitoring, menampilkan pesan "Update activity successfully!", dan menampilkan halaman detail monitoring
Alternative Flow: Jika aktor menekan tombol cancel pada saat menekan tombol End Activity	Sistem akan menutup formulir end activity dan menampilkan halaman detail monitoring.
Post-Condition	Sistem berhasil diperbarui dan menampilkan halaman detail monitoring

4.5.20 Melihat Daftar Rekomendasi Atlet

Use case scenario melihat daftar rekomendasi atlet dapat dilihat pada Tabel 4.22

Tabel 0.23 Use Case Scenario Melihat Daftar Rekomendasi Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar rekomendasi atlet.
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu <i>home</i> 2. Sistem akan menampilkan daftar rekomendasi atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar rekomendasi atlet dari tim

4.5.21 Melihat Daftar Atlet Hasil Monitoring Aktivitas Exercise

Use case scenario melihat daftar atlet hasil monitoring aktivitas exercise dapat dilihat pada tabel 4.23

Tabel 0.24 Use Case Scenario Melihat Daftar Atlet Hasil Monitoring Aktivitas Exercise

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan melihat daftar atlet hasil monitoring aktivitas exercise
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	1. Aktor menekan <i>home</i> pada menu 2. Sistem akan menampilkan melihat daftar atlet hasil monitoring aktivitas exercise
Alternative Flow	-
Post-Condition	Sistem menampilkan melihat daftar atlet hasil monitoring aktivitas exercise

4.5.22 Melihat Daftar Atlet Hasil Monitoring Aktivitas Sparing

Use case scenario melihat daftar atlet hasil monitoring aktivitas sparing dapat dilihat pada tabel 4.24

Tabel 0.25 Use Case Scenario Melihat Daftar Atlet Hasil Monitoring Aktivitas Sparing

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan melihat daftar atlet hasil monitoring aktivitas sparing
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	1. Aktor menekan menu <i>home</i> 2. Sistem akan menampilkan melihat daftar atlet hasil monitoring aktivitas sparing
Alternative Flow	-
Post-Condition	Sistem menampilkan melihat daftar atlet hasil monitoring aktivitas sparing

4.5.23 Melihat Daftar Atlet Hasil Monitoring Aktivitas Championship

Use case scenario melihat daftar atlet hasil monitoring aktivitas championship dapat dilihat pada tabel 4.25

Tabel 0.26 Use Case Scenario Melihat Daftar Atlet Hasil Monitoring Aktifitas Championship

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan melihat daftar atlet hasil monitoring aktifitas championship
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu <i>home</i> 2. Sistem akan menampilkan melihat daftar atlet hasil monitoring aktifitas championship
Alternative Flow	-
Post-Condition	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas championship

4.5.24 Melihat Daftar Tim

Use case scenario melihat daftar tim dapat dilihat pada tabel 4.26:

Tabel 0.27 Use Case Scenario Melihat Daftar Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar tim.
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan team pada header <i>dashboard</i> 2. Sistem akan menampilkan daftar tim
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar tim

4.5.25 Melihat Detail Tim

Use case scenario melihat detail tim dapat dilihat pada tabel 4.27

Tabel 0.28 Use Case Scenario Melihat Detail Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail tim.
Pre-Condition	Halaman menu berhasil di tampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan team pada menu 2. Sistem menampilkan data nama, deskripsi, avatar tim, daftar <i>achievement tim</i>, daftar atlet.
Alternative Flow: Jika parameter where null	Sistem akan menampilkan halaman 404 notfound

Post-Condition	Sistem menampilkan halaman detail tim.
-----------------------	--

4.5.26 Menambahkan Tim

Berikut merupakan *use case scenario* menambahkan tim dapat dilihat pada tabel 4.28

Tabel 0.29 Use Case Scenario Menambahkan Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menambahkan tim
Pre-Condition	Halaman detail tim berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol Add Team. 2. Sistem menampilkan formulir add team. 3. Aktor mengisi form add team berupa name, description, avatar, address, city, province, coach_id dan menekan tombol submit. 4. Sistem menyimpan data atlet tim dalam basis data, menampilkan pesan "Add team successfully!", dan menampilkan halaman detail tim
Alternative Flow: Jika name kurang dari 2 character atau lebih dari 191 character, dan / atau city lebih dari 191 character, dan/atau province lebih dari 191 character	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim
Post-Condition	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan daftar tim.

4.5.27 Mengubah Tim

Use case scenario mengubah tim dapat dilihat pada tabel 4.29

Tabel 0.30 Use Case Scenario Mengubah Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah tim
Pre-Condition	Halaman detail tim berhasil ditampilkan

Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit team. 2. Sistem menampilkan formulir mengubah tim. 3. Aktor mengisi form mengubah tim berupa name, description, avatar, address, city, province, coach_id dan menekan tombol submit. 4. Sistem menyimpan data tim ke dalam basis data dan menampilkan pesan "Update team successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan detail tim.

4.5.28 Menghapus Tim

Use case scenario menghapus tim dapat dilihat pada tabel 4.30

Tabel 0.31 Use Case Scenario Menghapus Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menghapus tim
Pre-Condition	Halaman detail tim berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol delete team. 2. Sistem menampilkan konfirmasi menghapus tim. 3. Aktor menekan tombol hapus. 4. Sistem menghapus data tim dari basis data dan menampilkan pesan "Delete team successfully!."
Alternative Flow	-
Post-Condition	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>

4.5.29 Melihat Daftar *Achievement* Tim

Use case scenario melihat daftar *achievement* tim dapat dilihat pada tabel 4.31

Tabel 0.32 Use Case Scenario Melihat Daftar *Achievement* Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar <i>achievement</i> tim.
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu team pada menu 2. Sistem akan menampilkan daftar <i>achievement</i> tim
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar <i>achievement</i> tim

4.5.30 Melihat Detail *Achievement* Tim

Use case scenario detail *achievement* tim dapat dilihat pada tabel 4.32

Tabel 0.33 *Use Case Scenario* Melihat Detail *Achievement* Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail <i>achievement</i> tim.
Pre-Condition	Halaman detail tim berhasil di tampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat pada tab <i>achievement</i> 2. Sistem akan menampilkan halaman detail <i>achievement</i> tim
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman detail <i>achievement</i> tim.

4.5.31 Menambahkan *Achievement* Tim

Use case scenario menambahkan *achievement* tim dapat d

Tabel 0.34 *Use Case Scenario* Menambahkan *Achievement* Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menambahkan <i>achievement</i> atlet
Pre-Condition	Halaman detail tim berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol add <i>achievement</i>. 2. Sistem menampilkan formulir tambah <i>achievement</i> tim. 3. Aktor mengisi form tambah <i>achievement</i> berupa name, description, image, level, date, team dan menekan tombol submit. 4. Sistem menyimpan data <i>achievement</i> tim ke dalam basis data dan menampilkan pesan "Add <i>achievement</i> successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> tim

4.5.32 Mengubah *Achievement* Tim

Use case scenario mengubah *achievement* tim dapat dilihat pada tabel 4.34

Tabel 0.35 *Use Case Scenario* Mengubah *Achievement* Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah <i>achievement</i> tim
Pre-Condition	Halaman detail <i>achievement</i> tim berhasil ditampilkan

Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit <i>achievement</i>. 2. Sistem menampilkan formulir mengubah <i>achievement</i> tim. 3. Aktor mengisi form mengubah <i>achievement</i> tim berupa name, description, image, level, date, team dan menekan tombol submit. 4. Sistem menyimpan data <i>achievement</i> tim ke dalam basis data dan menampilkan pesan "Update achievement successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> tim

4.5.33 Menghapus *Achievement* Tim

Use case scenario menghapus *achievement* Tim dapat dilihat pada tabel 4.35

Tabel 0.36 Use Case Scenario Menghapus *Achievement* Tim

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menghapus <i>achievement</i> tim
Pre-Condition	Halaman detail <i>achievement</i> tim berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol delete <i>achievement</i>. 2. Sistem menampilkan konfirmasi menghapus <i>achievement</i> tim. 3. Aktor menekan tombol "Yes". 4. Sistem menghapus data <i>achievement</i> tim dari basis data, menampilkan pesan "Delete <i>achievement</i> successfully!.", dan menampilkan halaman <i>dashboard</i>
Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> pelatih dan menampilkan halaman detail <i>achievement</i> pelatih
Post-Condition	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>

4.5.34 Melihat Daftar Atlet

Use case scenario melihat daftar atlet dapat dilihat pada tabel 4.36

Tabel 0.37 Use Case Scenario Melihat Daftar Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar atlet.

Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	1. Aktor menekan menu halaman atlet 2. Sistem akan menampilkan daftar atlet
Alternative Flow	-
Post-Condition	Daftar atlet berhasil ditampilkan

4.5.35 Melihat Detail Atlet

Use case scenario melihat detail atlet dapat dilihat pada tabel 4.37

Tabel 0.38 Use Case Scenario Melihat Detail Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail atlet.
Pre-Condition	Daftar atlet berhasil ditampilkan
Main Flow	1. Aktor menekan tombol lihat 2. Sistem menampilkan halaman data nama, jenis kelamin, avatar atlet, posisi atlet, daftar cedera, dan daftar <i>achievement</i> .
Alternative Flow	-
Post-Condition	Halaman detail atlet berhasil ditampilkan

4.5.36 Menambahkan Atlet

Use case scenario menambahkan atlet dapat dilihat pada tabel 4.38

Tabel 0.39 Use Case Scenario Menambahkan Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menambahkan atlet
Pre-Condition	Halaman tim berhasil ditampilkan
Main Flow	1. Aktor menekan tombol add <i>athlete</i> . 2. Sistem menampilkan formulir tambah atlet. 3. Aktor mengisi form tambah atlet berupa team, position, name, gender, avatar, address, date of birth, phone number, player number, status dan menekan tombol submit. 4. Sistem menyimpan data atlet ke dalam basis data dan menampilkan pesan "Add athlete successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar atlet.

4.5.37 Mengubah Atlet

Use case scenario mengubah atlet dapat dilihat pada tabel 4.39

Tabel 0.40 Use Case Scenario Mengubah Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah atlet
Pre-Condition	Halaman detail atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol Edit Athlete. 2. Sistem menampilkan formulir mengubah atlet. 3. Aktor mengisi form mengubah atlet berupa team, position, name, gender, avatar, address, date of birth, phone number, player number, status dan menekan tombol submit. 4. Sistem menyimpan data atlet ke dalam basis data dan menampilkan pesan "Update Athlete Successfully!."
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan halaman detail atlet.

4.5.38 Menghapus Atlet

Use case scenario menghapus atlet dapat dilihat pada tabel 4.40

Tabel 0.41 Use Case Scenario Menghapus Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menghapus atlet
Pre-Condition	Halaman detail atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol delete athlete. 2. Sistem menampilkan konfirmasi menghapus atlet. 3. Aktor menekan tombol "Yes". 4. Sistem menghapus data atlet dari basis data dan menampilkan pesan "Delete athlete successfully!."
Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus atlet dan menampilkan halaman detail atlet
Post-Condition	Sistem berhasil dihapus dan menampilkan halaman dashboard.

4.5.39 Melihat Daftar *Achievement* Atlet

Use case scenario melihat daftar *achievement* atlet dapat dilihat pada tabel 4.41

Tabel 0.42 Use Case Scenario Melihat Daftar *Achievement* Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar <i>achievement</i> atlet.
Pre-Condition	Halaman detail tim berhasil di tampilkan
Main Flow	1. Aktor menekan tombol lihat atlet 2. Sistem akan menampilkan daftar <i>achievement</i> atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar <i>achievement</i> atlet

4.5.40 Melihat Detail *Achievement* Atlet

Use case scenario detail *achievement* atlet dapat dilihat pada tabel 4.42

Tabel 0.43 Use Case Scenario Meliaht Detail *Achievement* Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail <i>achievement</i> atlet.
Pre-Condition	Halaman detail atlet berhasil di tampilkan
Main Flow	1. Aktor menekan tombol lihat pada tab <i>achievement</i> 2. Sistem akan menampilkan halaman detail <i>achievement</i> atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman detail <i>achievement</i> atlet.

4.5.41 Menambahkan *Achievement* Atlet

Use case scenario menambahkan *achievement* atlet dapat dilihat pada tabel 4.43

Tabel 0.44 Use Case Scenario Menambahkan *Achievement* Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menambahkan <i>achievement</i> atlet
Pre-Condition	Halaman detail atlet berhasil ditampilkan
Main Flow	1. Aktor menekan tombol add <i>achievement</i> atlet. 2. Sistem menampilkan formulir tambah <i>achievement</i> atlet. 3. Aktor mengisi form tambah <i>achievement</i> atlet berupa name, description, image, level, date, atlet dan menekan tombol submit.

	4. Sistem menyimpan data <i>achievement</i> atlet ke dalam basis data dan menampilkan pesan “Add <i>achievement</i> atlet successfully!”
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> atlet

4.5.42 Mengubah *Achievement* Atlet

Use case scenario mengubah *achievement* atlet dapat dilihat pada tabel 4.44

Tabel 0.45 Use Case Scenario Mengubah *Achievement* Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah <i>achievement</i> atlet
Pre-Condition	Halaman detail <i>achievement</i> atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit <i>achievement</i>. 2. Sistem menampilkan formulir mengubah <i>achievement</i> atlet. 3. Aktor mengisi form mengubah <i>achievement</i> atlet berupa name, description, image, level, date, atlet dan menekan tombol submit. 4. Sistem menyimpan data <i>achievement</i> atlet ke dalam basis data dan menampilkan pesan “Update <i>achievement</i> successfully!”
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> atlet

4.5.43 Menghapus *Achievement* Atlet

Use case scenario menghapus *achievement* atlet dapat dilihat pada tabel 4.45

Tabel 0.46 Use Case Scenario Menghapus *Achievement* Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menghapus <i>achievement</i> atlet
Pre-Condition	Halaman detail <i>achievement</i> atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol delete <i>achievement</i>. 2. Sistem menampilkan konfirmasi menghapus <i>achievement</i> atlet. 3. Aktor menekan tombol “Yes”. 4. Sistem menghapus data <i>achievement</i> atlet dari basis data dan menampilkan pesan “Delete <i>achievement</i> successfully!”, dan menampilkan halaman <i>dashboard</i>

Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> atlet dan menampilkan halaman detail <i>achievement</i> atlet
Post-Condition	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>

4.5.44 Melihat Daftar Riwayat Riwayat Cedera Atlet

Use case scenario melihat daftar riwayat cedera atlet dapat dilihat pada tabel 4.46

Tabel 0.47 Use Case Scenario Melihat Daftar Riwayat Cedera Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar riwayat cedera atlet.
Pre-Condition	Halaman detail tim berhasil di tampilkan
Main Flow	1. Aktor menekan tombol lihat atlet 2. Sistem akan menampilkan daftar riwayat cedera atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar riwayat cedera atlet

4.5.45 Melihat Detail Riwayat Cedera Atlet

Use case scenario melihat detail riwayat cedera atlet dapat dilihat pada tabel 4.47

Tabel 0.48 Use Case Scenario Melihat Detail Riwayat Cedera Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail riwayat cedera atlet.
Pre-Condition	Halaman detail atlet berhasil di tampilkan
Main Flow	1. Aktor menekan tombol lihat pada tab <i>history injury</i> 2. Sistem akan menampilkan halaman detail riwayat cedera atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman halaman detail riwayat cedera atlet.

4.5.46 Menambahkan Riwayat Cedera Atlet

Use case scenario menambahkan riwayat cedera atlet dapat dilihat pada tabel 4.48

Tabel 0.49 Use Case Scenario Menambahkan Riwayat Cedera Atlet

Action	Pelatih dan Admin
---------------	-------------------

Objective	Fungsi untuk dapat menambahkan riwayat cedera atlet
Pre-Condition	Halaman detail atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol Add injury. 2. Sistem menampilkan formulir tambah riwayat cedera atlet. 3. Aktor mengisi form tambah riwayat cedera atlet berupa name, description, avatar, date, athlete dan menekan tombol submit. 4. Sistem menyimpan data riwayat cedera atlet ke dalam basis data dan menampilkan pesan "Add history injury successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar riwayat cedera atlet.

4.5.47 Mengubah Riwayat Cedera Atlet

Use case scenario mengubah riwayat cedera atlet dapat dilihat pada tabel 4.49

Tabel 0.50 Use Case Scenario Mengubah Riwayat Cedera Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah riwayat cedera atlet
Pre-Condition	Halaman detail riwayat cedera atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit injury. 2. Sistem menampilkan formulir mengubah riwayat cedera atlet. 3. Aktor mengisi form mengubah riwayat cedera atlet berupa name, description, avatar, date, athlete dan menekan tombol submit. 4. Sistem menyimpan data riwayat cedera atlet ke dalam basis data dan menampilkan pesan "Update history injury successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan detail riwayat cedera atlet.

4.5.48 Menghapus Riwayat Cedera Atlet

Use case scenario menghapus riwayat cedera atlet dapat dilihat pada tabel 4.50

Tabel 0.51 Use Case Scenario Menghapus Riwayat Cedera Atlet

Action	Pelatih dan Admin
---------------	-------------------

Objective	Fungsi untuk dapat menghapus riwayat cedera atlet
Pre-Condition	Halaman atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol delete injury. 2. Sistem menampilkan konfirmasi menghapus riwayat cedera atlet. 3. Aktor menekan tombol "Yes". 4. Sistem menghapus data riwayat cedera atlet dari basis data, menampilkan pesan "Delete history injury successfully!.", dan menampilkan halaman <i>dashboard</i>
Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus riwayat cedera atlet dan menampilkan halaman detail riwayat cedera atlet
Post-Condition	Riwayat cedera atlet berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>

4.5.49 Melihat Daftar Aktifitas Atlet

Use case scenario melihat daftar aktifitas atlet dapat dilihat pada tabel 4.51

Tabel 0.52 Use Case Scenario Melihat Daftar Aktifitas Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk menampilkan daftar aktifitas atlet
Pre-Condition	Halaman detail tim berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat detail atlet 2. Sistem akan menampilkan daftar aktifitas atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar aktifitas atlet

4.5.50 Melihat Detail Parameter Penilaian

Use case scenario melihat detail parameter penilaian dapat dilihat pada tabel 4.52

Tabel 0.53 Use Case Scenario Melihat Detail Parameter Penilaian

Action	Pelatih dan Admin
Objective	Fungsi untuk melihat detail parameter penilaian.
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu parameters pada menu 2. Sistem menampilkan halaman detail parameter penilaian
Alternative Flow	-

Post-Condition	Sistem menampilkan halaman detail parameter penilaian
-----------------------	---

4.5.51 Melihat Daftar Parameter Tetap Penilaian Atlet

Use case scenario melihat daftar parameter tetap penilaian atlet dapat dilihat pada tabel 4.53

Tabel 0.54 Use Case Scenario Melihat Daftar Parameter Tetap Penilaian Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat melihat daftar parameter tetap penilaian atlet
Pre-Condition	Halaman parameter penilaian berhasil ditampilkan
Main Flow	1. Aktor menekan menekan tab <i>fixed parameter</i> . 2. Sistem akan menampilkan daftar parameter tetap penilaian atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar parameter tetap penilaian atlet

4.5.52 Melihat Daftar Parameter Tidak Tetap Penilaian Atlet

Use case scenario melihat daftar parameter tidak tetap penilaian atlet dapat dilihat pada tabel 4.54

Tabel 0.55 Use Case Scenario Melihat Daftar Parameter Tidak Tetap Penilaian Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat melihat daftar parameter tidak tetap penilaian atlet
Pre-Condition	Halaman parameter penilaian berhasil ditampilkan
Main Flow	1. Aktor menekan menekan tab <i>your parameter</i> . 2. Sistem akan menampilkan daftar parameter tidak tetap penilaian atlet
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar parameter tetap penilaian atlet

4.5.53 Menambahkan Parameter Tidak Tetap Penilaian Atlet

Use case scenario menambahkan parameter tidak tetap penilaian atlet dapat dilihat pada tabel 4.55

Tabel 0.56 Use Case Scenario Menambahkan Parameter Tidak Tetap Penilaian Atlet

Action	Pelatih dan Admin
---------------	-------------------

Objective	Fungsi untuk dapat menambahkan parameter tidak tetap penilaian atlet
Pre-Condition	Halaman parameter tidak tetap penilaian atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol Tambah. 2. Sistem menampilkan formulir tambah parameter tidak tetap penilaian atlet. 3. Aktor mengisi form parameter tidak tetap penilaian atlet berupa name, short name, type dan menekan tombol submit. 4. Sistem menyimpan data parameter tidak tetap penilaian atlet dalam basis data dan menampilkan pesan "Add parameters successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar parameter tidak tetap penilaian atlet

4.5.54 Mengubah Parameter Tidak Tetap Penilaian Atlet

Use case scenario mengubah parameter tidak tetap penilaian atlet dapat dilihat pada tabel 4.56

Tabel 0.57 Use Case Scenario Mengubah Parameter Tidak Tetap Penilaian Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah parameter tidak tetap penilaian atlet
Pre-Condition	Halaman parameter tidak tetap penilaian atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit. 2. Sistem menampilkan formulir mengubah parameter tidak tetap penilaian atlet. 3. Aktor mengisi form mengubah parameter tidak tetap penilaian atlet berupa name, short name, type dan menekan tombol submit. 4. Sistem menyimpan data parameter tidak tetap penilaian atlet ke dalam basis data dan menampilkan pesan "Update parameters successfully!"
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan daftar parameter tidak tetap penilaian atlet

4.5.55 Menghapus Parameter Tidak Tetap Penilaian Atlet

Use case scenario menghapus parameter tidak tetap penilaian atlet dapat dilihat pada tabel 4.57

Tabel 0.58 Use Case Scenario Menghapus Parameter Tidak Tetap Penilaian Atlet

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat menghapus parameter tidak tetap penilaian atlet
Pre-Condition	Halaman parameter tidak tetap penilaian atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>delete</i>. 2. Sistem menampilkan konfirmasi menghapus parameter tidak tetap penilaian atlet. 3. Aktor menekan tombol "Yes". 4. Sistem menghapus data parameter tidak tetap penilaian atlet dari basis data dan menampilkan pesan "Delete parameters successfully!"
Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus parameter tidak tetap penilaian atlet dan menampilkan daftar parameter tidak tetap penilaian atlet
Post-Condition	Sistem berhasil dihapus dan menampilkan daftar parameter tidak tetap penilaian atlet

4.5.56 Mengubah Persentase Tipe Parameter Penilaian

Use case scenario mengubah persentase tipe parameter penilaian dapat dilihat pada tabel 4.58

Tabel 0.59 Use Case Scenario Mengubah Persentase Tipe Parameter Penilaian

Action	Pelatih dan Admin
Objective	Fungsi untuk dapat mengubah persentase tipe parameter penilaian
Pre-Condition	Halaman detail parameter penilaian berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit percentage. 2. Sistem menampilkan formulir mengubah persentase parameter penilaian. 3. Aktor mengubah data persentase parameter dengan total penjumlahan penilaian lebih dari 100 atau kurang dari 100, dan menekan tombol simpan

	4. Sistem menyimpan data persentase parameter penilaian ke dalam basis data dan menampilkan pesan "Update percentage successfully!"
Alternative Flow: Jika data total persentase parameter penilaian kurang dari atau lebih dari 100%	Sistem akan menampilkan pesan error dan menampilkan halaman detail parameter penilaian
Post-Condition	Sistem berhasil mengubah dan menampilkan detail parameter penilaian

4.5.57 Melihat Daftar Cabang Olahraga

Use case scenario melihat daftar cabang olahraga dapat dilihat pada tabel 4.59

Tabel 0.60 Use Case Scenario Melihat Daftar Cabang Olahraga

Action	Admin
Objective	Fungsi untuk menampilkan daftar cabang olahraga.
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	1. Aktor menekan menu halaman cabang olahraga 2. Sistem akan menampilkan daftar cabang olahraga
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar cabang olahraga

4.5.58 Melihat Detail Cabang Olahraga

Use case scenario melihat detail cabang olahraga dapat dilihat pada tabel 4.60

Tabel 0.61 Use Case Scenario Melihat Detail Cabang Olahraga

Action	Admin
Objective	Fungsi untuk melihat cabang olahraga.
Pre-Condition	Daftar cabang olahraga berhasil ditampilkan
Main Flow	1. Aktor menekan tombol lihat 2. Sistem menampilkan data nama cabang olahraga dan deskripsi cabang olahraga.
Alternative Flow	-
Post-Condition	Sistem menampilkan informasi halaman detail cabang olahraga.

4.5.59 Menambahkan Cabang Olahraga

Use case scenario menambahkan cabang olahraga dapat dilihat pada tabel 4.61

Tabel 0.62 Use Case Scenario Menambahkan Cabang Olahraga

Action	Admin
Objective	Fungsi untuk dapat menambahkan cabang olahraga
Pre-Condition	Halaman parameter penilaian atlet berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol Tambah. 2. Sistem menampilkan formulir tambah cabang olahraga. 3. Aktor mengisi form tambah cabang olahraga berupa name, description dan menekan tombol submit. 4. Sistem menyimpan data cabang olahraga dalam basis data, menampilkan pesan "Data berhasil disimpan.", dan menampilkan daftar cabang olahraga
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar cabang olahraga

4.5.60 Mengubah Cabang Olahraga

Use case scenario mengubah cabang olahraga dapat dilihat pada tabel 4.62

Tabel 0.63 Use Case Scenario Mengubah Cabang Olahraga

Action	Admin
Objective	Fungsi untuk dapat mengubah cabang olahraga
Pre-Condition	Halaman cabang olahraga berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit. 2. Sistem menampilkan formulir mengubah cabang olahraga. 3. Aktor mengisi form mengubah data cabang olahraga berupa name, description dan menekan tombol submit. 4. Sistem menyimpan data cabang olahraga ke dalam basis data, menampilkan pesan "Data berhasil disimpan.", dan menampilkan daftar cabang olahraga
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan daftar cabang olahraga

4.5.61 Menghapus Cabang Olahraga

Use case scenario menghapus cabang olahraga dapat dilihat pada tabel 4.63

Tabel 0.64 Use Case Scenario Menghapus Cabang Olahraga

Action	Admin
Objective	Fungsi untuk dapat menghapus cabang olahraga
Pre-Condition	Halaman cabang olahraga berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol hapus. 2. Sistem menampilkan konfirmasi menghapus cabang olahraga. 3. Aktor menekan tombol "Yes". 4. Sistem menghapus data cabang olahraga dari basis data, menampilkan pesan "Data berhasil dihapus.", dan menampilkan daftar cabang olahraga
Alternative Flow: Jika aktor menekan tombol Cancel	Sistem akan menutup form konfirmasi menghapus cabang olahraga dan menampilkan daftar cabang olahraga.
Post-Condition	Sistem berhasil dihapus dan menampilkan daftar cabang olahraga

4.5.62 Melihat Daftar Pelatih

Use case scenario melihat daftar pelatih dapat dilihat pada tabel 4.64

Tabel 0.65 Use Case Scenario Melihat Daftar Pelatih

Action	Admin
Objective	Fungsi untuk menampilkan daftar pelatih.
Pre-Condition	Halaman menu berhasil ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu halaman pelatih 2. Sistem akan menampilkan daftar pelatih
Alternative Flow	-
Post-Condition	Sistem menampilkan daftar pelatih.

4.5.63 Melihat Detail Pelatih

Use case scenario melihat detail pelatih dapat dilihat pada tabel 4.65

Tabel 0.66 Use Case Scenario Melihat Detail Pelatih

Action	Admin
Objective	Fungsi untuk melihat pelatih.

Pre-Condition	Daftar pelatih berhasil ditampilkan
Main Flow	1. Aktor menekan tombol lihat 2. Sistem menampilkan data nama, avatar, email, olahraga, dan daftar tim.
Alternative Flow	-
Post-Condition	Sistem menampilkan detail pelatih.

4.5.64 Menambahkan Pelatih

Use case scenario menambahkan pelatih dapat dilihat pada tabel 4.66

Tabel 0.67 Use Case Scenario Menambahkan Pelatih

Action	Admin
Objective	Fungsi untuk dapat menambahkan pelatih
Pre-Condition	Halaman daftar pelatih berhasil ditampilkan
Main Flow	1. Aktor menekan tombol Tambah. 2. Sistem menampilkan formulir tambah pelatih. 3. Aktor mengisi data pelatih dan menekan tombol submit. 4. Sistem menyimpan data pelatih dalam basis data dan menampilkan pesan "Data berhasil disimpan."
Alternative Flow	-
Post-Condition	Sistem berhasil ditambahkan dan menampilkan daftar pelatih

4.5.65 Mengubah pelatih

Use case scenario mengubah pelatih dapat dilihat pada tabel 4.67

Tabel 0.68 Use Case Scenario Mengubah Pelatih

Action	Admin
Objective	Fungsi untuk dapat mengubah pelatih
Pre-Condition	Halaman daftar pelatih berhasil ditampilkan
Main Flow	1. Aktor menekan tombol edit. 2. Sistem menampilkan formulir mengubah pelatih. 3. Aktor mengubah data pelatih. 4. Sistem menyimpan data pelatih ke dalam basis data dan menampilkan pesan "Data berhasil disimpan."
Alternative Flow	-
Post-Condition	Sistem berhasil diperbarui dan menampilkan daftar pelatih

4.5.66 Menghapus Pelatih

Use case scenario menghapus pelatih dapat dilihat pada tabel 4.68

Tabel 0.69 Use Case Scenario Menghapus Pelatih

Action	Admin
Objective	Fungsi untuk dapat menghapus pelatih
Pre-Condition	Halaman pelatih berhasil ditampilkan
Main Flow	<ol style="list-style-type: none">1. Aktor menekan tombol hapus.2. Sistem menampilkan konfirmasi menghapus pelatih.3. Aktor menekan tombol yes.4. Sistem menghapus data pelatih dari basis data, menampilkan pesan "Data berhasil dihapus.", dan menampilkan daftar pelatih
Alternative Flow	-
Post-Condition	Sistem berhasil dihapus dan menampilkan daftar pelatih



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Tahapan penelitian selanjutnya adalah perancangan. Hasil analisis kebutuhan yang telah dijelaskan pada bab sebelumnya digunakan sebagai standar perancangan. Pada bab ini dilakukan identifikasi yang mengacu pada spesifikasi kebutuhan dan *use case scenario*. pada bab ini juga akan dijabarkan secara detail mengenai analisis kebutuhan dan akibatnya terhadap perancangan. Perancangan perangkat dibagi menjadi 5 tahapan, yaitu pemodelan *Sequence Diagram*, pemodelan *Class Diagram*, perancangan Algoritma, perancangan Basis data, dan perancangan Antarmuka.

5.1.1 Pemodelan *Sequence Diagram*

Sequence Diagram merupakan diagram yang mendeskripsikan kolaborasi antar objek. *Use Case Scenario* yang terlampir pada hasil analisis kebutuhan dimanfaatkan untuk merancang pemodelan *Sequence Diagram*. *Sequence Diagram* dirancang pada setiap *Use Case Scenario* sebanyak 64 *Sequence Diagram*. Namun hanya 3 *Sequence Diagram* yang ditunjukkan sebagai sampel, yaitu *Sequence Diagram* menambahkan tim, melihat detail tim, dan mengubah score atlet. Ketiganya merupakan fungsi utama yang dipilih pada hasil analisis kebutuhan dan akan dimanfaatkan sampai tahap pengujian.

1. *Sequence Diagram* Menambahkan Tim

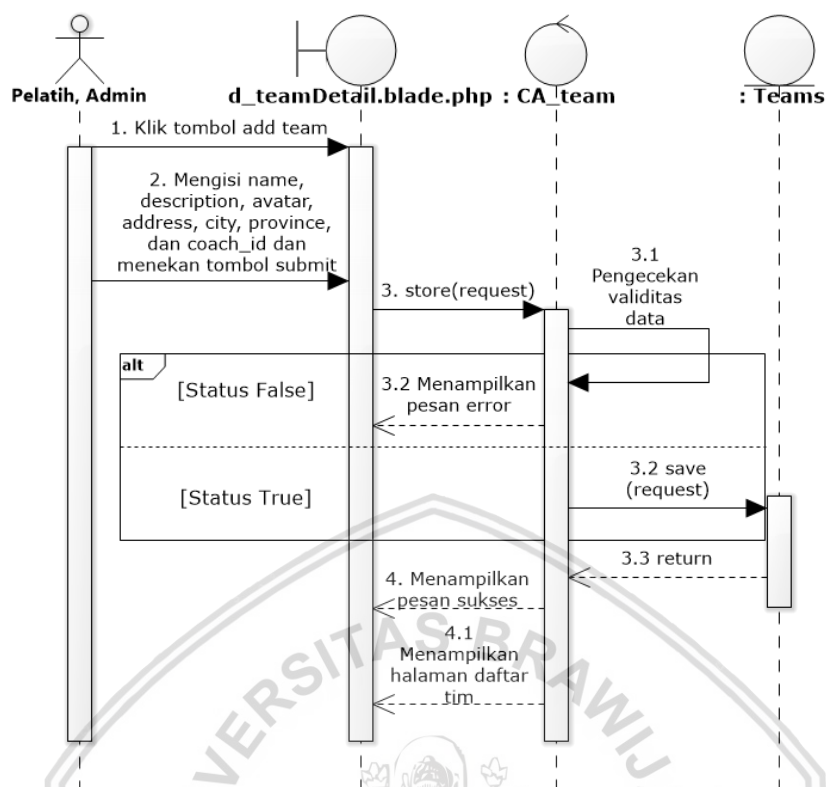
Sequence Diagram menambahkan tim menjelaskan alur yang terjadi pada saat melakukan menambahkan tim ke dalam sistem. Selengkapnya dapat dilihat pada Gambar 5.1.

2. *Sequence Diagram* Melihat Detail Tim

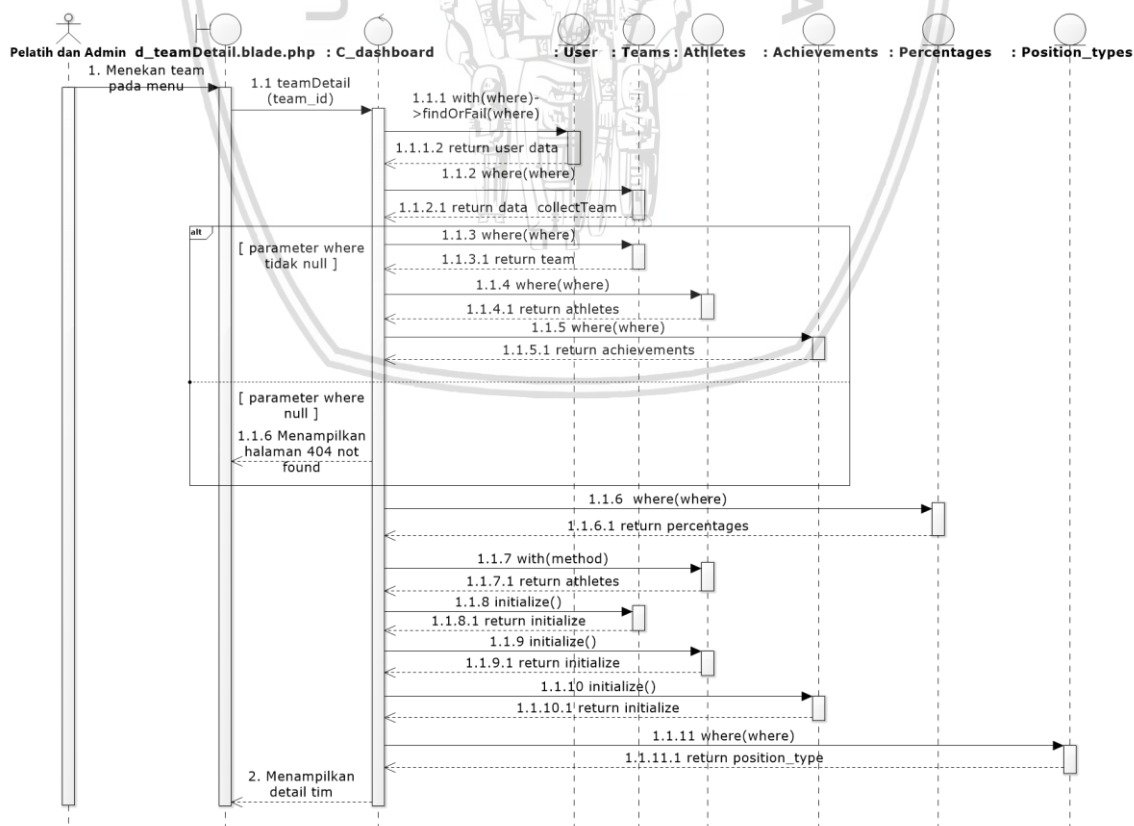
Sequence Diagram melihat detail tim menjelaskan alur yang terjadi pada saat memilih menu team. Selengkapnya dapat dilihat pada Gambar 5.2.

3. *Sequence Diagram* Mengubah Score Atlet

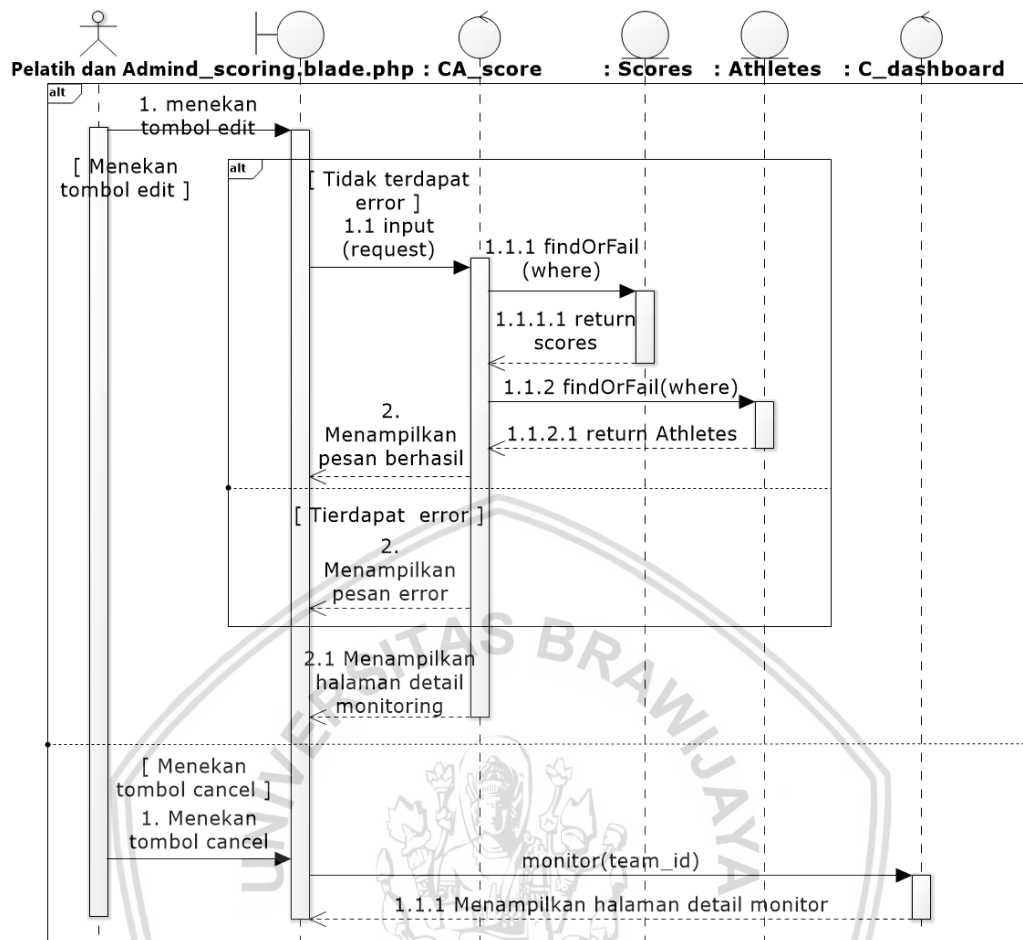
Sequence Diagram Mengubah Score Atlet menjelaskan alur yang terjadi pada saat menekan tombol edit pada halaman detail monitoring. Selengkapnya dapat dilihat pada Gambar 5.3.



Gambar 0.1 Sequence Diagram Menambahkan Tim



Gambar 0.2 Sequence Diagram Melihat Detail Tim

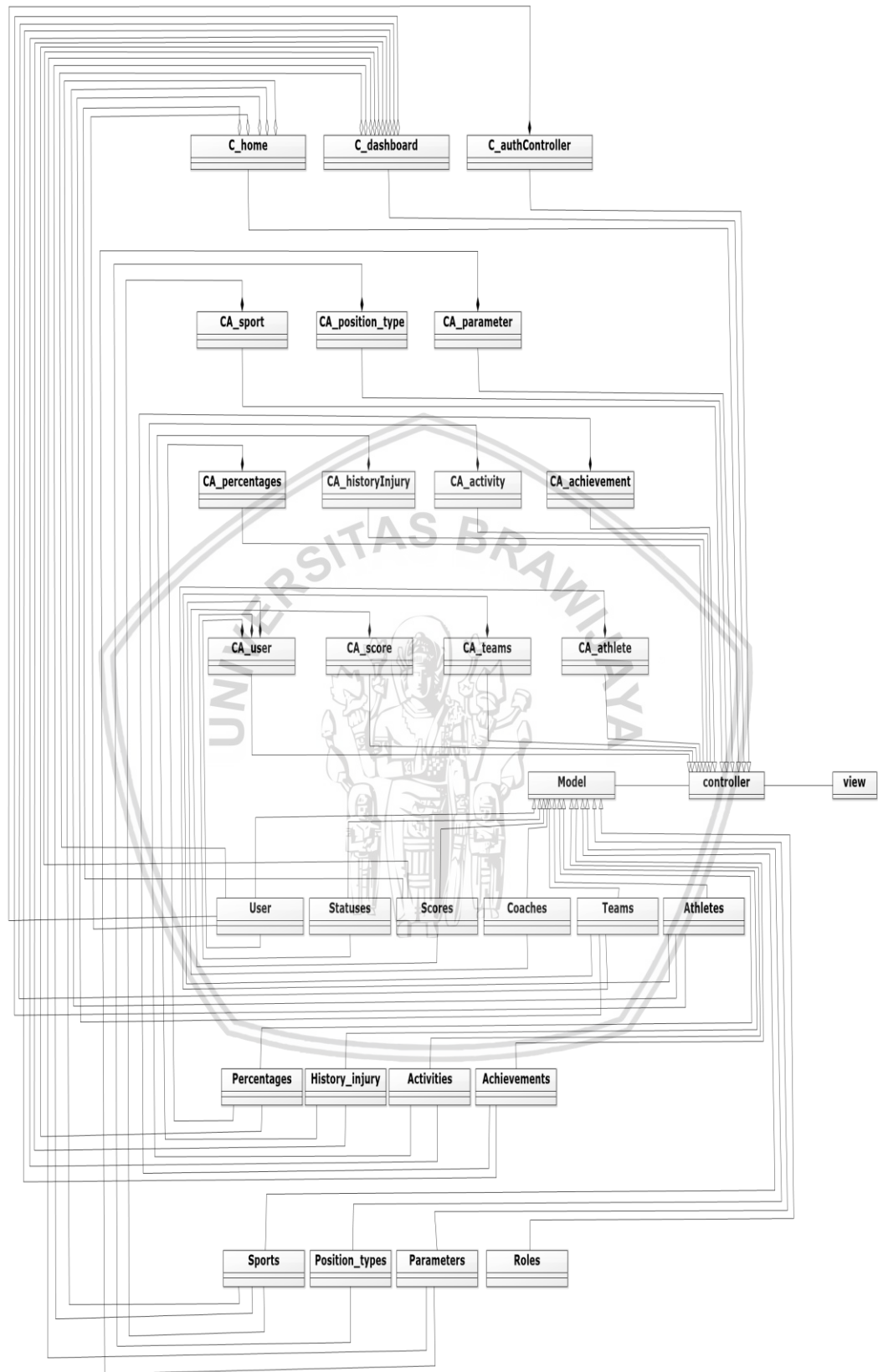


Gambar 0.3 Sequence Diagram Mengubah Score Atlet

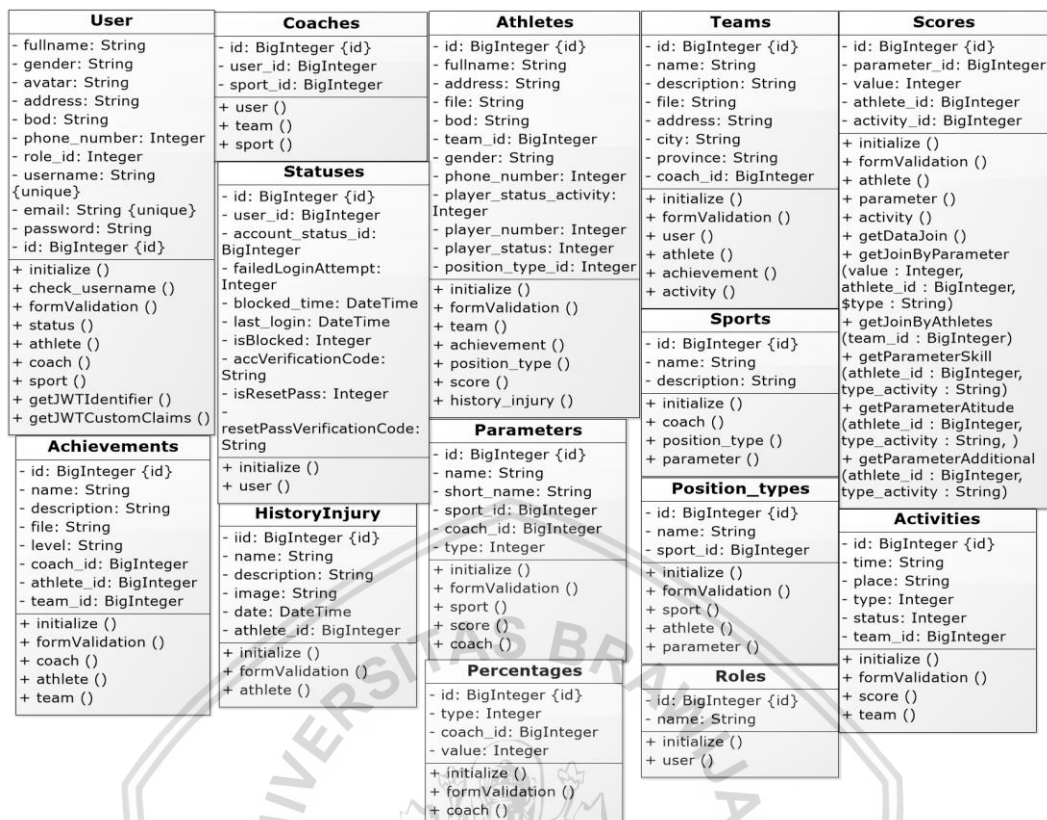
5.1.2 Pemodelan Class Diagram

Setelah pemodelan *Sequence Diagram* dirancang, berikutnya peneliti merancang pemodelan *Class Diagram*. Rancangan *Sequence Diagram* digunakan dalam perancangan pemodelan *Class Diagram*. Pemodelan *Class Diagram* berpatokan pada *framework* MVC David Naista. *Class-class* yang akan dikategorikan menjadi 3 *class* utama, antara lain; *class Model*, *class View*, dan *class Controller* dan 2 *class* pendukung, yakni *class Database* dan *class ResultSet* dimana merupakan *library*.

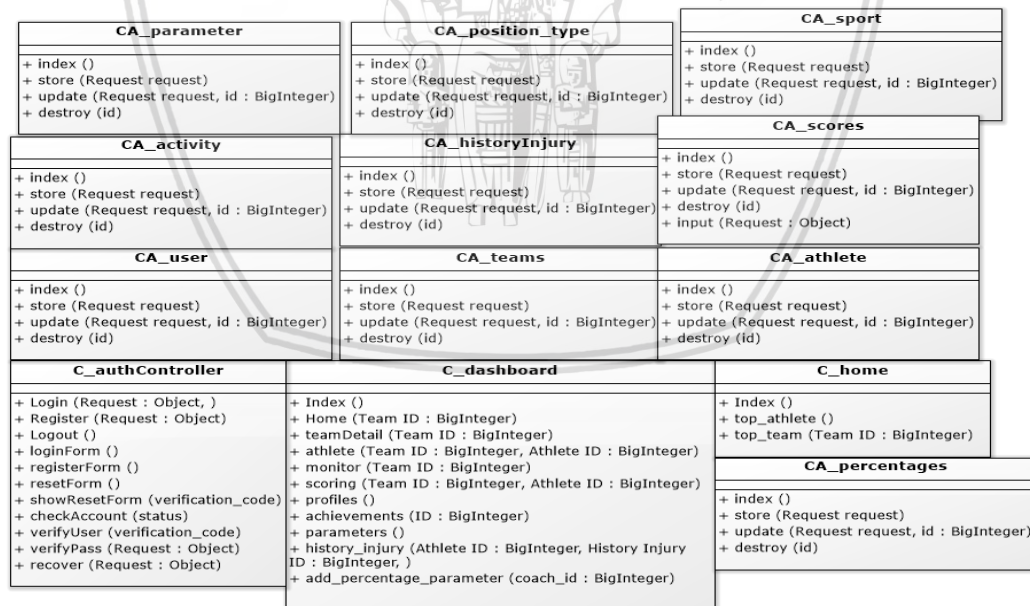
Pada *class Model* terdapat method untuk menginisialisasi model dan basis data dan method untuk mengeksekusi *query* secara manual. Pada *class View* terdapat method yang sering digunakan untuk mengakses halaman *view*. Pada *Class Controller* terdapat method umum yang digunakan *class Controller* untuk menginisialisasi model dan memanggil *view*. Pada *class Database* terdapat method yang berguna sebagai pengelola basis data. Secara detail terdapat pada Gambar 5.6.



Gambar 0.4 *Class Diagram* Perancangan Umum



Gambar 0.5 Class Diagram pada Model



Gambar 0.6 Class Diagram Controller

Dari Gambar 5.6, terdapat 13 *class* memiliki hubungan inheritance dengan *class* Controller serta 14 *class* memiliki hubungan inheritance dengan *class* Model. Pada *class* controller terdapat fungsi *template()* yang berfungsi untuk memanggil *class* statis yang berisi halaman view dimana selalu tampil pada setiap halaman dinamis.

1.1.1 Perancangan Algoritme

Setelah perancangan *Class Diagram* telah dilakukan, berikutnya dijelaskan algoritma setiap operasi pada setiap *class*. Algoritma setiap operasi akan menjelaskan bagaimana runtutan langkah suatu perangkat untuk dapat menghasilkan *output*. Namun hanya ada 3 operasi dari beberapa *class* yang akan dijabarkan sebagai sampel, antara lain; *store (request)* pada *class CA_team*, operasi *add()* pada *class TimController*, operasi *team_detail(team_id)* pada *class C_dashboard*, dan operasi *input(request)* pada *class CA_score*. Dibawah ini merupakan penjelasan setiap algoritma:

1. Algoritme operasi store(request) klas CA_team

Berikut penjelasan algoritme operasi score (request) pada klas CA_team :

1	Deklarasi variabel roles
2	WHILE terdapat hasil pemanggilan method formValidation dengan dijadikan variabel key
3	IF request[key] tidak kosong
4	roles[key] = request[key]
5	END IF
6	END WHILE
7	Inisiasi variabel validator = memanggil validator method dengan parameter roles dan melakukan pengecekan data
8	IF validator = false
9	Menampilkan pesan error
10	END IF
11	IF request tidak memiliki file bertipe gambar
12	Menampilkan pesan error
13	END IF
14	Inisiasi variabel team = instansiasi model Team() Memasukan nilai pada variabel team sebagai object (
15	name = request[name],
16	description = request[description],
17	address = request[address],
18	city = request[city],
19	province = request[province],
20	coach_id = request[coach_id]
)
21	Menyimpan data ke dalam basis data dengan memanggil method save() pada instansiasi team
22	Inisialisasi variabel file = request[file]
23	IF folder 'public/images/team/[team id]' belum tersedia
24	Membuat folder tersebut
25	END IF
26	Inisialisasi variabel path = membuat path folder untuk menyimpan gambar berdasarkan inputan request[file]
27	path = memotong string '/' agar menjadi sebuah array

28	inisialisasi variabel insertFile = instansiasi model Teams
29	Memasukan nilai pada variabel team sebagai object (
30	avatar = memasukan potongan2 path yang diperlukan
31)
32	Menyimpan data ke dalam basis data dengan memanggil method save() pada instansiasi insertFile
33	Menampilkan pesan sukses

2. Algoritme operasi teamDetail(team_id) klas C_dashboard

Berikut penjelasan algoritme operasi teamDetail(team_id) pada klas C_dashboard :

1	Mulai
2	Inisiasi variabel data = memasukan nilai dari object (
3	title = 'User Dashboard',
4	parent_nav = 'team',
5	child_nav' => 'detail',
6	header_title => ''
7)
8	Inisiasi variabel session = mengambil session
9	Inisiasi variabel user = mendapat kembalian dari memanggil method with(where)->findOrFail(where) dari model User
10	Inisiasi variabel coach = mendapat kembalian dari memanggil method find(where) dari model coach
11	Inisiasi variabel collectTeam = memanggil method where(where)->get() dari model Teams
12	TRY
13	Inisiasi variabel team = mendapat kembalian dari memanggil method findOrFail(where)->first() dari model Teams
14	Inisiasi variabel athlete = mendapat kembalian dari memanggil method where(athlete)->get() dari model Athletes
15	Inisiasi variabel achievement = mendapat kembalian dari memanggil method where(athlete)->get() dari model Achievements
16	JIKA TERJADI ERROR
17	Menampilkan halaman error 404 notfound
18	END TRY
19	Inisiasi variabel percentages = mendapat kembalian dari memanggil method where(where)->get() dari model Percentages
20	IF perncetages == kosong
21	Melakukan pemanggilan method
22	add_percentage_parametercoach_ud);
23	END IF
24	Inisiasi variabel collectAthlete
25	Inisiasi variabel count = 0;
26	FOR menjabarkan object athlete sebagai variabel key

	collectAthlete[\$count++] = mendapat kembalian dari memanggil
27	method with(where)->findOrFail(where) dari model Athletes
28	END FOR
29	Inisiasi variabel form_team = mendapat kembalian dari memanggil
30	method initialize() dari model Teams
	Inisiasi variabel form_athlete = mendapat kembalian dari memanggil
31	method initialize() dari model Athletes
	Inisiasi variabel form_achievement = mendapat kembalian dari memanggil
32	method initialize() dari model Achievements
	Inisiasi variabel position = = mendapat kembalian dari memanggil method
33	find(where) dari model Position_types
34	Data[header_title] = team[name];
	menampilkan halaman detail tim

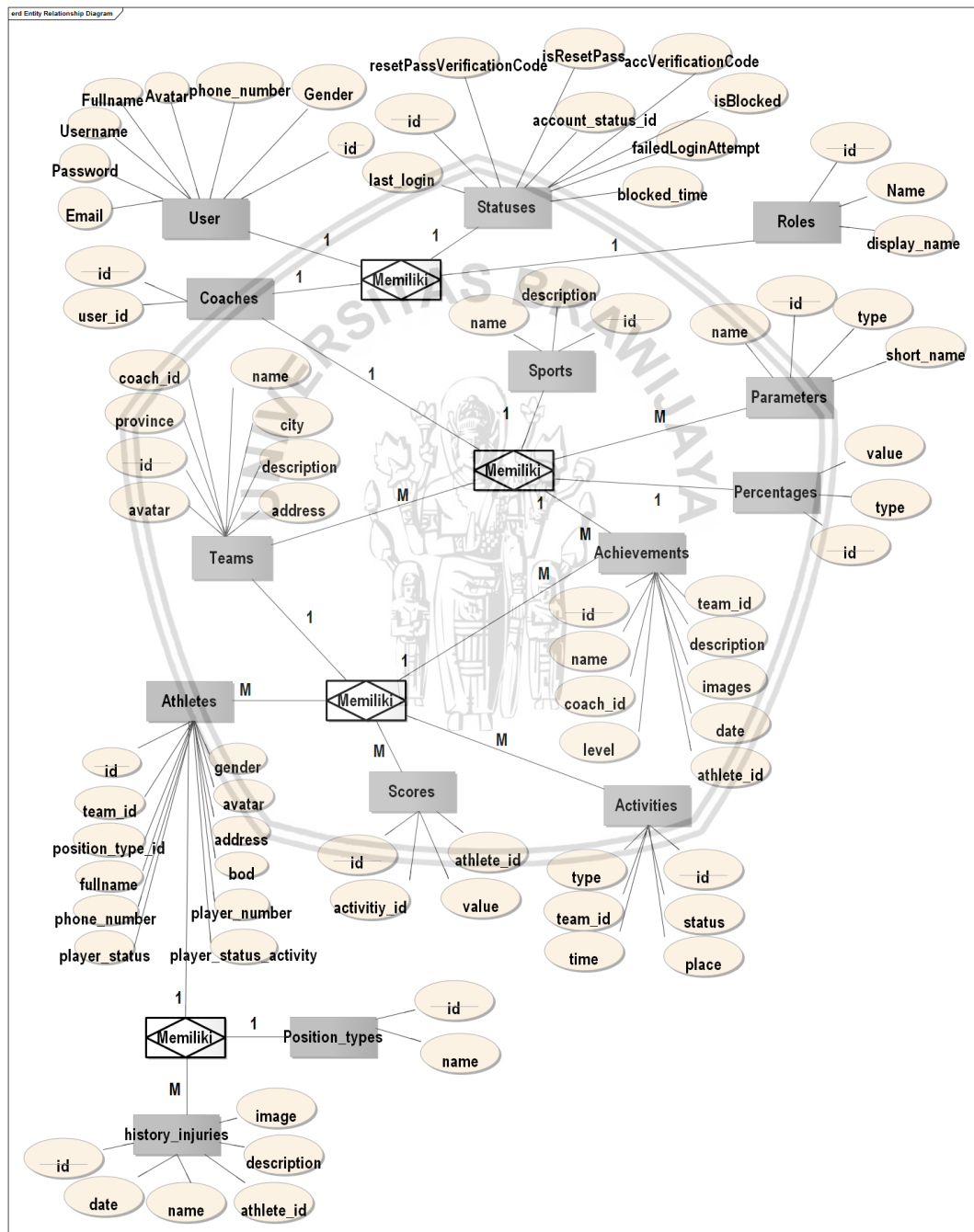
3. Algoritme operasi input(request) klas CA_score

Berikut penjelasan algoritme operasi input(request) pada klas CA_score:

1	Mulai
2	Inisiasi variabel score = menghilangkan index terakhir dari request[score] score = memecah variabel score ke dalam sebuah array berdasarkan string ‘ , ‘
3	Inisiasi variabel value = menghilangkan index terakhir dari request[value]
4	value = memecah variabel value ke dalam sebuah array berdasarkan string ‘ , ‘
5	TRY
6	FOR index i = 0, i < panjang array score, i++
7	Inisiasi variabel update = mendapat balikan dari memanggil method findOrFail(where) pada model Scores
8	Memasukan nilai pada variabel update sebagai object (
9	value = value[i]
10)
11	Menyimpan object tersebut kedalam basis data dengan memanggil method save()
12	END FOR
13	Inisiasi variabel athlete = mendapat balikan dari memanggil method findOrFail(where) pada model Athletes
14	Memasukan nilai pada variabel athlete sebagai object (
15	player_status_activity = “done”
16)
17	Menyimpan object tersebut kedalam basis data dengan memanggil method save()
18	Menampilkan pesan berhasil
19	JIKA TERJADI ERROR
20	Menampilkan pesan error
21	END TRY

1.1.2 Perancangan Basis Data

Perancangan basis data akan dikerjakan secara terkonsep dan logis. Perancangan basis data yang dikerjakan dengan terkonsep akan membuat *Entity Relationship Diagram* (ER Diagram) yang mana hanya menampilkan Nama Entitas dan Hubungan antar Entitas. Secara detail pada Gambar 5.7



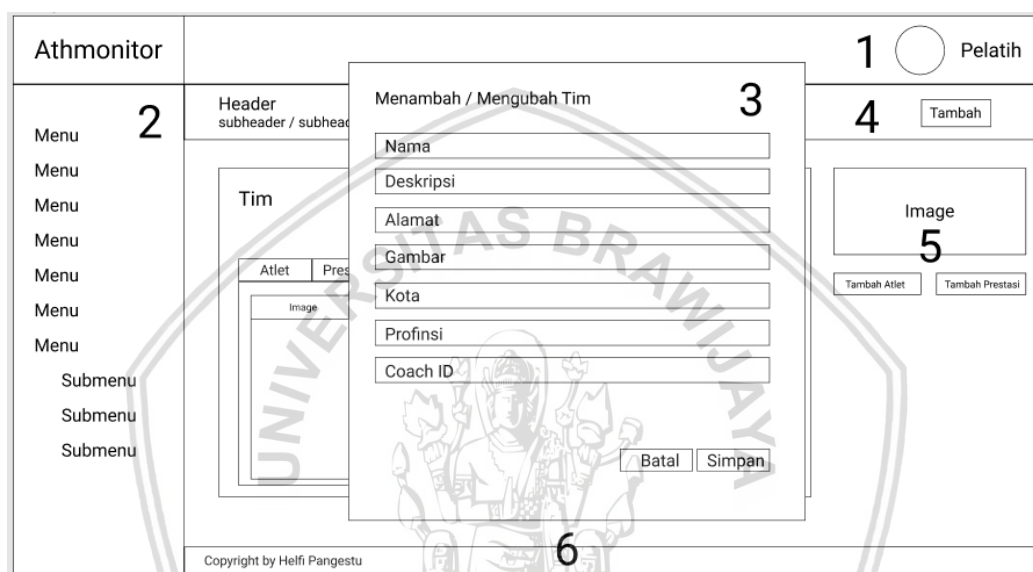
Gambar 0.7 Entity Relational Diagram

1.1.3 Perancangan Antarmuka

Perancangan antarmuka adalah pembuatan rancangan tampilan yang akan menjadi jembatan antara perangkat yang akan dibuat dengan pengguna. Namun hanya 3 rancangan antarmuka yang dijelaskan sebagai sampel, yaitu antarmuka menambahkan tim, melihat detail tim, dan mengubah score atlet. Berikut adalah masing-masing rancangan antarmuka tersebut:

1. Rancangan Antarmuka Menambahkan Tim

Hasil perancangan antarmuka menambahkan tim dapat dilihat pada Gambar 5.8 dan akan dijelaskan pada Tabel 5.1.



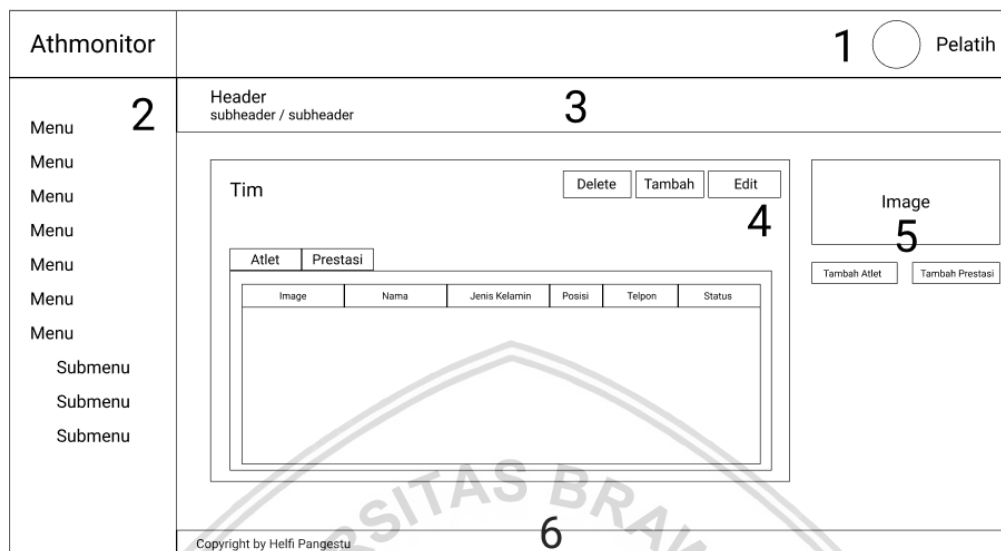
Gambar 0.8 Rancangan Antarmuka Menambahkan Tim

Tabel 0.1 Penjelasan Rancangan Antarmuka Menambahkan Tim

No	Nama Objek	Keterangan
1	Header	Menampilkan header dari sistem yang berisi <i>user icon</i> dan nama lengkap pengguna yang dapat diklik.
2	Sidebar Kiri	Berisi menu-menu yang tersedia pada sistem
3	Formulir Tim	Formulir untuk menambah atau mengubah tim
4	Subheader	Menampilkan nama subheader dan tombol untuk menambah tim
5	Sidebar Kanan	Menampilkan gambar dari tim dan tombol untuk menambah atlet dan menambah prestasi tim
6	Footer	Menampilkan footer dari sistem

2. Rancangan Antarmuka Melihat Detail Tim

Hasil perancangan antarmuka melihat detail tim dapat dilihat pada Gambar 5.9 dan akan dijelaskan pada Tabel 5.2.



Gambar 0.9 Rancangan Antarmuka Melihat Detail Tim

Tabel 0.2 Penjelasan Rancangan Antarmuka Melihat Detail Tim

No	Nama Objek	Keterangan
1	Header	Menampilkan header dari sistem yang berisi <i>user icon</i> dan nama lengkap pengguna yang dapat diklik.
2	Sidebar Kiri	Berisi menu-menu yang tersedia pada sistem
3	Subheader	Menampilkan nama subheader dan tombol untuk menambah tim
4	Daftar Tim	Menampilkan daftar tim, tombol untuk menambah tim, mengubah tim, dan menghapus tim
5	Sidebar Kanan	Menampilkan foto tim, tombol menambah atlet, dan menambah <i>achievement</i>
6	Footer	Menampilkan footer dari sistem

3. Rancangan Antarmuka Mengubah Score Atlet

Hasil perancangan antarmuka mengubah score atlet dapat dilihat pada Gambar 5.10 dan akan dijelaskan pada Tabel 5.3

The wireframe shows a web application layout. At the top right is a header area (1) containing a circular user icon and the text 'Pelatih'. On the left is a sidebar menu (2) with items: Menu, Menu, Menu, Menu, Menu, Menu, Submenu, Submenu, and Submenu. The main content area (3) has a subheader and a list of parameters (4): Parameter, Attitude, Skill, and Additional. Each parameter has a corresponding slider control. At the bottom right of the main area are 'Cancel' and 'Submit' buttons. The footer (5) contains the text 'Copyright by Hefli Pangestu'.

Gambar 0.10 Rancangan Antarmuka Mengubah Score Atlet

Tabel 0.3 Penjelasan Rancangan Antarmuka Mengubah Score Atlet

No	Nama Objek	Keterangan
1	Header	Menampilkan header dari sistem yang berisi <i>user icon</i> dan nama lengkap pengguna yang dapat diklik.
2	Sidebar Kiri	Berisi menu-menu yang tersedia pada sistem
3	Subheader	Menampilkan nama subheader dan tombol untuk menambah tim
4	Daftar parameter scoring	Menampilkan daftar parameter penilaian bertipe attitude, skill, dan additional serta terdapat tombol submit untuk melakukan penyimpanan dan tombol cancel untuk menggagalkan aktifitas
5	Footer	Menampilkan footer dari sistem

1.2 Implementasi

Setelah melakukan tahap perancangan, tahap selanjutnya ialah tahap implementasi. Proses implementasi mengacu dari hasil perancangan. Setiap hasil perancangan akan diimplementasikan pada sistem. Pada tahap implementasi, akan menjelaskan spesifikasi sistem dalam perangkat keras dan perangkat lunak, implementasi basis data, dan implementasi antarmuka dan hasil.

1.2.1 Spesifikasi Sistem

Dalam tahap implementasi, proses ini didukung oleh perangkat keras dan perangkat lunak. Berikut akan dijabarkan secara jelas spesifikasi dari perangkat lunak dan perangkat keras yang digunakan dalam proses implementasi.

a. Spesifikasi Perangkat Keras

Berikut spesifikasi dari perangkat keras, dapat dilihat pada Tabel 5.16:

Tabel 0.4 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	Intel(R) Core(TM) i5-4210U CPU @ 1.70Ghz, 2.4GHz
RAM	8 GB
<i>Harddisk</i>	1 TB

b. Spesifikasi Perangkat Lunak

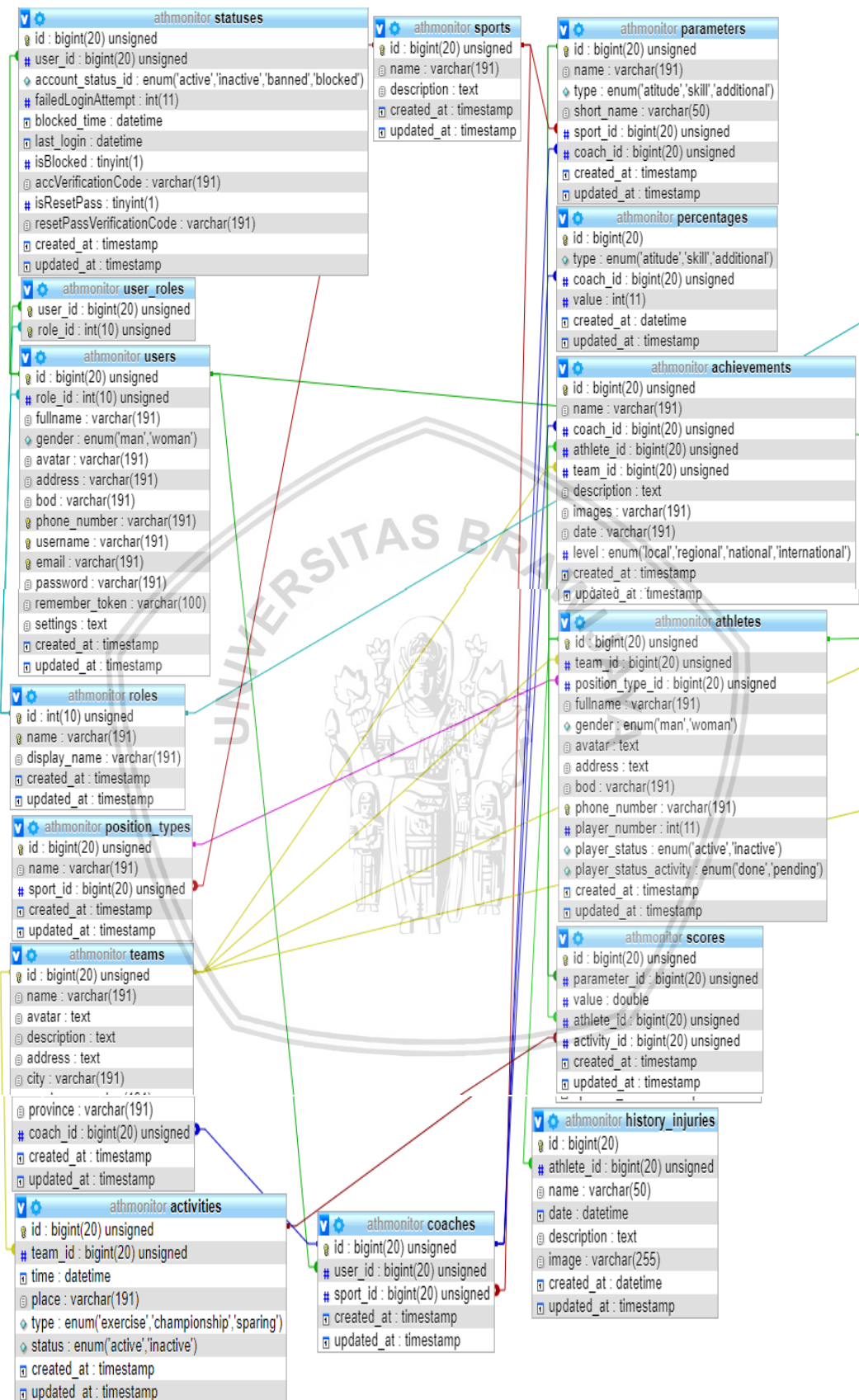
Berikut spesifikasi dari perangkat lunak, dapat dilihat pada Tabel 5.17:

Tabel 0.5 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemrograman	HTML, CSS, JavaScript, PHP, dan SQL
<i>Framework</i>	Bootstrap dan MVC

1.2.2 Implementasi Basis Data

Implementasi basis data dilakukan berdasarkan hasil perancangan basis data yang ada pada tahap perancangan. *Schema* basis data yang digunakan adalah MariaDB dan hasil implementasi basis data digambarkan pada Gambar 5.11.



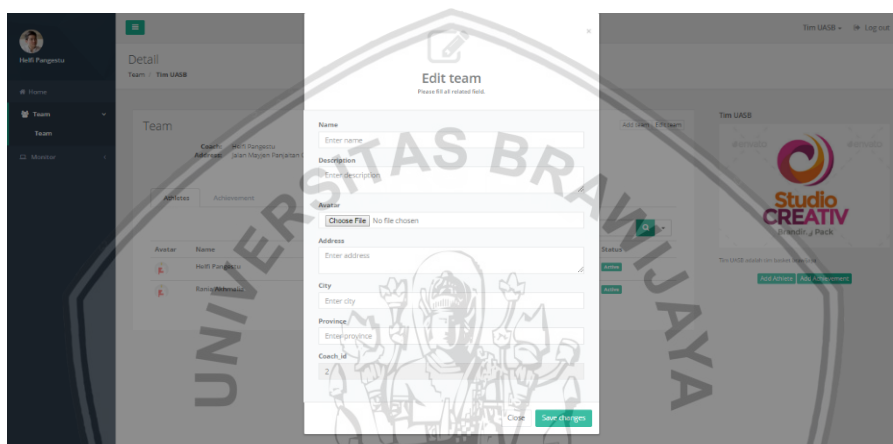
Gambar 0.11 Hasil Implementasi Basis Data

5.2.3 Implementasi Antarmuka

Standar yang digunakan pada proses implementasi adalah hasil perancangan antarmuka yang telah dijabarkan pada sub-bab perancangan. Sampel implementasi antarmuka yang akan dijelaskan adalah 3 tampilan antarmuka yang telah ditentukan pada perancangan antarmuka, yaitu antarmuka menambahkan tim, melihat detail tim, dan mengubah score atlet. Berikut tampilan masing-masing antarmuka tersebut:

1. Tampilan Antarmuka Menambahkan Tim

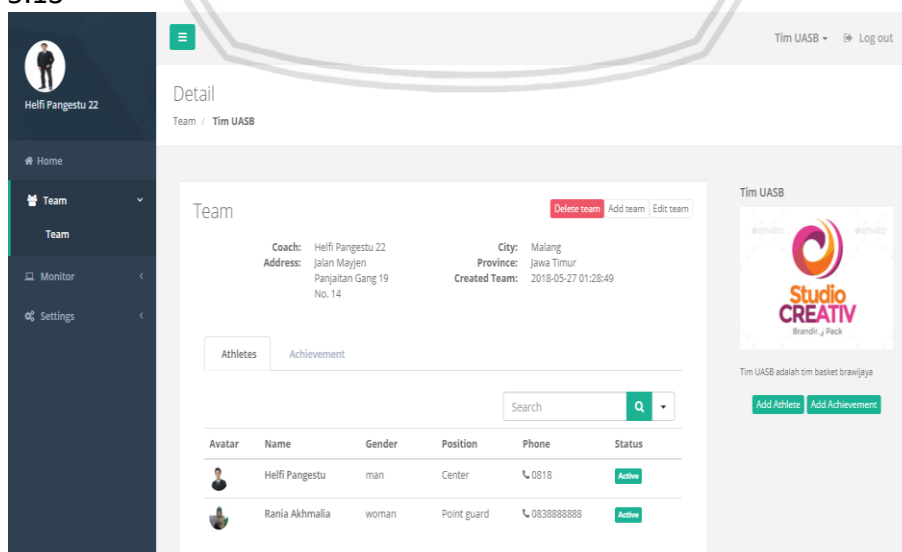
Hasil implementasi antarmuka menambahkan tim dapat dilihat pada Gambar 5.12.



Gambar 0.12 Tampilan Antarmuka Menambahkan Tim

2. Tampilan Antarmuka Melihat Detail Tim

Hasil implementasi antarmuka melihat detail tim dapat dilihat pada Gambar 5.13



Gambar 0.13 Tampilan Antarmuka Melihat Detail Tim

3. Tampilan Antarmuka Mengubah Score Atlet

Hasil implementasi antarmuka mengubah score atlet dapat dilihat pada Gambar 5.14

The screenshot shows a web application interface for monitoring and updating athlete scores. The interface is divided into a sidebar and a main content area.

Sidebar:

- Home
- Team
- Monitor (selected)
- Settings

Main Content Area:

Monitor / Scoring

Parameter Please enter all related data.

Attitude

- Attitude
- On Time
- Teamwork

Skill

- 3 Point Field Goals Attempted
- 3 Point Field Goals Made
- Assists
- Blocks
- Defensive Rebounds
- Efficiency
- Field Goals Attempted
- Field Goals Made
- Free Throws Attempted
- Free Throws Made
- Games Played
- Minutes Played
- Offensive Rebounds
- Points
- Rebounds
- Steals
- Testing 1
- Turnovers

Additional

- Absensi
- Tidak Hadir
- TP1

Buttons: Cancel, Submit

Gambar 0.14 Tampilan Antarmuka Mengubah Score Atlet

BAB 6 PENGUJIAN

6.1 Pengujian Unit

Pengujian unit adalah pengujian untuk menguji komponen/unit yang ada di dalam *source code*. Pengujian ini dilakukan untuk memastikan algoritme yang telah diimplementasikan telah sesuai atau tidak terhadap analisis kebutuhan dan perancangan sistem yang dibuat. Pengujian ini menggunakan *basis path testing* sebagai teknik pengujian.

6.1.1 Pengujian Unit Algoritma Menambahkan Tim

1. *Pseudocode Algoritme* Menambahkan Tim

Berikut merupakan algoritme dari operasi store(request) klas CA_team:

Tabel 0.1 *Pseudocode Algoritme* Menambahkan Tim

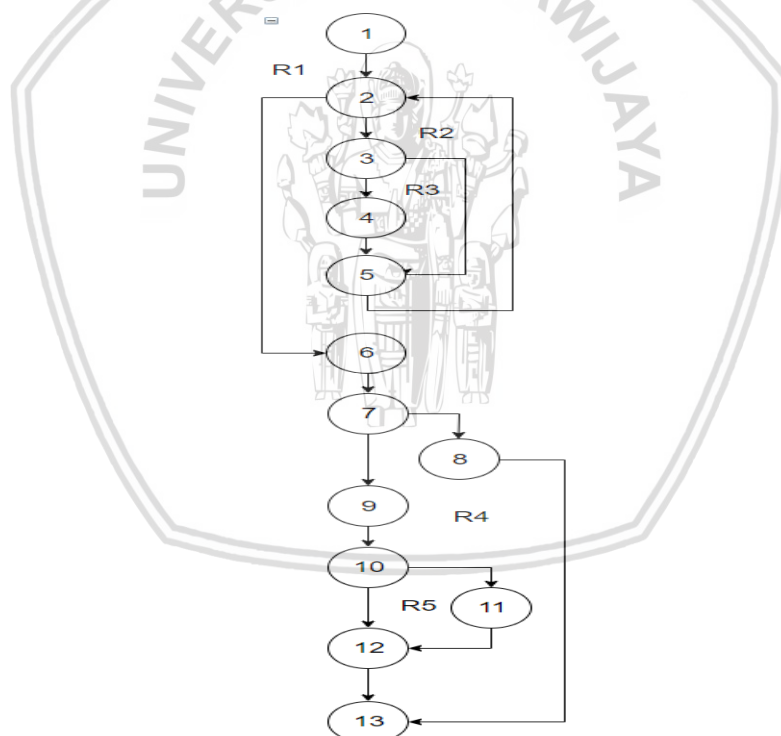
1	Mulai Deklarasi variabel roles
2	FOR terdapat hasil pemanggilan method formValidation dengan dijadikan variabel key
3	IF request[key] tidak kosong
4	roles[key] = request[key]
5	END IF
6	END FOR Inisiasi variabel validator = memanggil validator method dengan parameter roles dan melakukan pengecekan data
7	IF validator = false
8	Menampilkan pesan error
9	END IF Inisiasi variabel team = instansiasi model Team() Memasukan nilai pada variabel team sebagai object (name = request[name], description = request[description], address = request[address], city = request[city], province = request[province], coach_id = request[coach_id]) Menyimpan data ke dalam basis data dengan memanggil method save() pada instansiasi team Inisialisasi variabel file = request[file]
10	IF folder 'public/images/team/[team id]' belum tersedia
11	Membuat folder tersebut
12	END IF

	Inisialisasi variabel path = membuat path folder untuk menyimpan gambar berdasarkan inputan request[file] path = memotong string '/' agar menjadi sebuah array inisialisasi variabel insertFile = instansiasi model Teams Memasukan nilai pada variabel team sebagai object (avatar = memasukan potongan2 path yang diperlukan) Menyimpan data ke dalam basis data dengan memanggil method save() pada instansiasi insertFile Menampilkan pesan sukses
13	Selesai

2. Basis Path Testing

2.1. Flow graph

Gambar 6.1 merupakan *flow graph* dari algoritme operasi store(request) klas CA_team:



Gambar 0.1 Flow Graph Menambahkan Tim

2.2. Cyclomatic complexity

Berikut merupakan perhitungan *cyclomatic complexity* dari *flow graph* operasi store(request) klas CA_team:

- 1) $V(G) : \text{Total Ruang. Total } R = 5$
- 2) $V(G) : \text{Jumlah Panah} - \text{Jumlah Node} + 2 = 15 - 13 + 2 = 5$

3) $V(G) : \text{Total Percabangan (P)} + 1$. Total $P = 4 + 1 = 5$

2.3. Independent path

Berikut merupakan jalur-jalur *independent path* dari *flow graph* operasi *store(request)* klas *CA_team*:

- 1) Jalur 1 = 1 – 2 – 6 – 7 – 8 – 13
- 2) Jalur 2 = 1 – 2 – 3 – 5 – 2 – 6 – 7 – 8 – 13
- 3) Jalur 3 = 1 – 2 – 3 – 4 – 5 – 2 – 6 – 7 – 8 – 13
- 4) Jalur 4 = 1 – 2 – 3 – 4 – 5 – 2 – 6 – 7 – 9 – 10 – 12 – 13
- 5) Jalur 5 = 1 – 2 – 3 – 4 – 5 – 2 – 6 – 7 – 9 – 10 – 11 – 12 – 13

2.4. Hasil Pengujian

Tabel 6.2 merupakan hasil pengujian dari operasi *store(request)* klas *CA_team* berdasarkan jalur *independent path* yang dihasilkan:

Tabel 0.2 Kasus Uji Algoritme Menambahkan Tim

No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Penguji mengisi form <i>name, description, avatar, address, city, province</i> , dan <i>coach_id</i> dengan valid, dan <i>formValidation</i> = kosong	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim	Valid
2	Penguji mengisi form <i>name, description, avatar, address, city, province</i> , dan <i>coach_id</i> dengan valid namun key pada perulangan tidak sesuai saat melakukan percabangan <i>request[key] != kosong</i>	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim	Valid
3	Penguji mengisi <i>name</i> = kurang dari 2	Sistem akan menampilkan	Sistem akan menampilkan	Valid

	character atau lebih dari 191 character, dan / atau city = lebih dari 191 character , dan/atau province = lebih dari 191 character	pesan error dan menampilkan halaman detail tim	n pesan error dan menampilkan halaman detail tim	
4	Penguji mengisi form name, description, avatar, address , city, province, dan coach_id dengan valid dan sesuai dengan variabel key pada perulangan <i>formValidation</i> dan folder 'public/images/team/[team id]' tersedia	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan daftar tim.	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan daftar tim.	Valid
5	Penguji mengisi form name, description, avatar, address , city, province, dan coach_id dengan valid dan sesuai dengan variabel key pada perulangan <i>formValidation</i> dan folder 'public/images/team/[team id]' belum tersedia	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan daftar tim.	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan daftar tim.	Valid

6.1.2 Pengujian Unit Algoritma Melihat Detail Tim

1. *Pseudocode algoritme* melihat detail tim

Berikut merupakan algoritme dari operasi teamDetail(team_id) klas C_dashboard:

Tabel 0.3 Pseudocode Algoritme Melihat Detail Tim

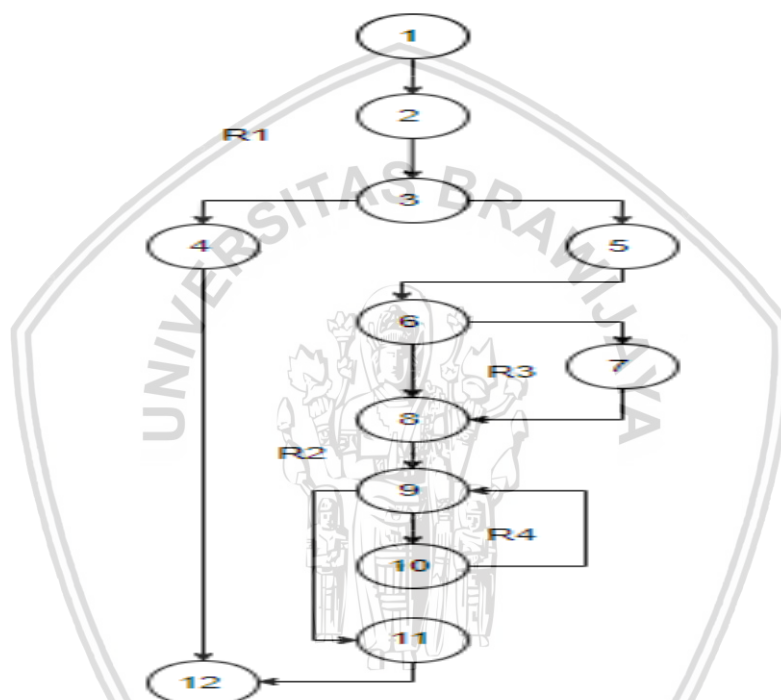
1	<p>Mulai</p> <p>Inisiasi variabel data = memasukan nilai dari object (</p> <p> title = 'User Dashboard',</p> <p> parent_nav = 'team',</p> <p> child_nav' => 'detail',</p> <p> header_title => ''</p> <p>)</p> <p>Inisiasi variabel session = mengambil session</p> <p>Inisiasi variabel user = mendapat kembalian dari memanggil method with(where)->findOrFail(where) dari model User</p> <p>Inisiasi variabel coach = mendapat kembalian dari memanggil method find(where) dari model coach</p> <p>Inisiasi variabel collectTeam = memanggil method where(where)->get() dari model Teams</p>
2	TRY
3	<p> Inisiasi variabel team = mendapat kembalian dari memanggil method findOrFail(where)->first() dari model Teams</p> <p> Inisiasi variabel athlete = mendapat kembalian dari memanggil method where(athlete)->get() dari model Athletes</p> <p> Inisiasi variabel achievement = mendapat kembalian dari memanggil method where(athlete)->get() dari model Achievements</p>
4	<p> JIKA TERJADI ERROR</p> <p> Menampilkan halaman error 404 notfound</p>
5	<p>END TRY</p> <p>Inisiasi variabel percentages = mendapat kembalian dari memanggil method where(where)->get() dari model Percentages</p>
6	IF percentages == kosong
7	Melakukan pemanggilan method add_percentage_parameter(coach_id)
8	<p>END IF</p> <p>Inisiasi variabel collectAthlete</p> <p>Inisiasi variabel count = 0</p>
9	FOR menjabarkan object athlete sebagai variabel key
10	collectAthlete[\$count++] = mendapat kembalian dari memanggil method with(where)->findOrFail(where) dari model Athletes
11	<p>END FOR</p> <p>Inisiasi variabel form_team = mendapat kembalian dari memanggil method initialize() dari model Teams</p> <p>Inisiasi variabel form_athlete = mendapat kembalian dari memanggil method initialize() dari model Athletes</p> <p>Inisiasi variabel form_achievement = mendapat kembalian dari memanggil method initialize() dari model Achievements</p>

	Inisiasi variabel position = = mendapat kembalian dari memanggil method find(wher) dari model Position_types Data[header_title] = team[name]; menampilkan halaman detail tim
12	selesai

2. Basis Path Testing

2.1. Flow graph

Gambar 6.2 merupakan *flow graph* dari algoritme operasi teamDetail(team_id) klas C_dashboard:



Gambar 0.2 Flow graph Melihat Detail Tim

2.2. Cyclomatic complexity

Berikut merupakan perhitungan *cyclomatic complexity* dari *flow graph* operasi teamDetail(team_id) klas C_dashboard:

- 1) $V(G) : \text{Total Ruang. Total } R = 4$
- 2) $V(G) : \text{Jumlah Panah} - \text{Jumlah Node} + 2 = 14 - 12 + 2 = 4$
- 3) $V(G) : \text{Total Percabangan (P)} + 1. \text{ Total } P = 3 + 1 = 4$

2.3. Independent path

Berikut merupakan jalur-jalur *independent path* dari *flow graph* operasi teamDetail(team_id) klas C_dashboard:

- 1) Jalur 1 = 1 – 2 – 3 – 4 – 12
- 2) Jalur 2 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12

3) Jalur 3 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 11 – 12

4) Jalur 4 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 9 – 11 – 12

2.4. Hasil Pengujian

Tabel 6.4 merupakan hasil pengujian dari operasi teamDetail(team_id) klas C_dashboard berdasarkan jalur independent path yang dihasilkan:

Tabel 0.4 Kasus Uji Algoritme Melihat Detail Tim

No Jalur	Prosedur Uji	<i>Expected Result</i>	<i>Result</i>	Status
1	variabel team = kosong	Sistem menampilkan halaman 404 notfound	Sistem menampilkan halaman 404 notfound	Valid
2	Variabel percentages = tidak kosong, Variabel athlete = kosong	Sistem menampilkan halaman detail tim dengan tabel atlet yang kosong.	Sistem menampilkan halaman detail tim dengan tabel atlet yang kosong.	Valid
3	Variabel percentages = kosong, Variabel athlete = kosong	Sistem menampilkan halaman detail tim dengan tabel atlet yang kosong.	Sistem menampilkan halaman detail tim dengan tabel atlet yang kosong.	Valid
4	Variabel percentages = kosong, Variabel athlete = tidak kosong	Sistem menampilkan halaman detail tim dengan tabel atlet yang berisi.	Sistem menampilkan halaman detail tim dengan tabel atlet yang berisi.	Valid

6.13 Pengujian Unit Algoritma Mengubah Score Atlet

1. Pseudocode Algoritme Mengubah Score Atlet

Berikut merupakan *pseudocode algoritme* dari operasi input(request) pada klas CA_score:

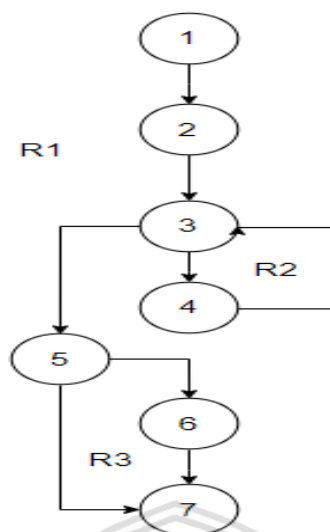
Tabel 0.5 Pseudocode Algoritme Mengubah Score Atlet

1	Mulai Inisiasi variabel score = menghilangkan index terakhir dari request[score] score = memecah variabel score ke dalam sebuah array berdasarkan string ' , ' Inisiasi variabel value = menghilangkan index terakhir dari request[value] value = memecah variabel value ke dalam sebuah array berdasarkan string ' , '
2	TRY
3	FOR index i = 0, i < panjang array score, i++
4	Inisiasi variabel update = mendapat balikan dari memanggil method findOrfail(wher) pada model Scores Memasukan nilai pada variabel update sebagai object (value = value[i]) Menyimpan object tersebut kedalam basis data dengan memanggil method save()
5	END FOR Inisiasi variabel athlete = mendapat balikan dari memanggil method findOrfail(wher) pada model Athletes Memasukan nilai pada variabel athlete sebagai object (player_status_activity = "done") Menyimpan object tersebut kedalam basis data dengan memanggil method save() Menampilkan pesan berhasil
6	JIKA TERJADI ERROR Menampilkan pesan error
7	END TRY Selesai

2. Basis Path Testing

2.1. Flow graph

Gambar 6.3 merupakan flow graph dari algoritme operasi input(request) pada klas CA_score:



Gambar 0.3 Flow graph Mengubah Score Atlet

2.2. Cyclomatic complexity

Berikut merupakan perhitungan cyclomatic complexity dari flow graph operasi input(request) pada klas CA_score:

- 1) $V(G)$: Total Ruang. Total $R = 3$
- 2) $V(G)$: Jumlah Panah – Jumlah Node + 2 = $8 - 7 + 2 = 3$
- 3) $V(G)$: Total Percabangan (P) + 1. Total $P = 2 + 1 = 3$

2.3. Independent path

Berikut merupakan jalur-jalur independent path dari flow graph operasi input(request) pada klas CA_score:

- 1) Jalur 1 = 1 – 2 – 3 – 5 – 7
- 2) Jalur 2 = 1 – 2 – 3 – 4 – 3 – 5 – 7
- 3) Jalur 3 = 1 – 2 – 3 – 4 – 3 – 5 – 6 – 7

2.4. Hasil Pengujian

Tabel 6.6 merupakan hasil pengujian dari operasi input(request) pada klas CA_score berdasarkan jalur independent path yang dihasilkan:

Tabel 0.6 Kasus Uji Algoritme Mengubah Score Atlet

No Jalur	Prosedur Uji	Expected Result	Result	Status
1	variabel score = 0	Sistem berhasil diperbarui namun tidak ada score yang	Sistem berhasil diperbarui namun tidak ada score yang	Valid

		disimpan, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.	disimpan, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.	
2	variabel score > 0	Sistem berhasil diperbarui, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.	Sistem berhasil diperbarui, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.	Valid
3	variabel score > 0, variabel athlete = null	Sistem akan menampilkan pesan error dan menampilkan halaman scoring	Sistem akan menampilkan pesan error dan menampilkan halaman scoring	Valid

6.1.4 Hasil Kasus Uji

Pengujian pada kasus uji yang pertama memiliki nilai *cyclomatic complexity* bernilai 4, kasus uji yang kedua memiliki nilai *cyclomatic complexity* bernilai 4, dan kasus uji yang ketiga memiliki nilai *cyclomatic complexity* bernilai 3. Nilai *cyclomatic complexity* pada tiga algoritme yang diuji termasuk dalam rentang nilai 1-10. Menurut (Guru99, 2018), rentang nilai 1-10 berarti sistem yang dibangun memiliki kode struktur yang baik dan high testability.

6.2 Pengujian Validasi

Dalam pengujian sistem, pengujian dilakukan untuk mengetahui apakah sistem yang telah dikembangkan telah sesuai dengan setiap kebutuhan yang telah ditentukan pada tahap analisis kebutuhan. Pengujian yang dilakukan menggunakan *blackbox testing*. Dalam pengujian validasi ini akan dilakukan pemeriksaan kesesuaian kebutuhan fungsional yang telah didefinisikan sebelumnya.

6.2.1 Pengujian Login

Pengujian sitem untuk *login* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_01_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Login

Tabel 0.7 Kasus Uji Berhasil Login

Nomor Kasus Uji	MA_01_00
Nama Kasus Uji	Login
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol Login pada halaman <i>home</i> 2. Aktor mengisi username dan password dengan benar 3. Aktor mengklik tombol login.
Hasil yang diharapkan	Sistem berhasil melakukan autentikasi aktor dan menampilkan halaman utama aktor
Hasil	Sistem berhasil melakukan autentikasi actor dan menampilkan halaman utama aktor
Status	Valid

2. Kasus Uji Login Alternatif 1: Jika data *username* dan *password* tidak valid

Tabel 0.8 Kasus Uji Login Alternatif 1

Nomor Kasus Uji	MA_01_00
Nama Kasus Uji	Kasus Uji Login Alternatif 1: Jika data <i>username</i> dan <i>password</i> tidak valid
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol Login pada halaman <i>home</i> 2. Aktor mengisi username dan password dengan salah 3. Aktor mengklik tombol login.
Hasil yang diharapkan	Sistem akan menampilkan pesan error dan menampilkan halaman utama pengunjung
Hasil	Sistem akan menampilkan pesan error dan menampilkan halaman utama pengunjung
Status	Valid

3. Kasus Uji Login Alternatif 2: Jika data *username* dan *password* kosong.

Tabel 0.9 Kasus Uji Login Alternatif 2

Nomor Kasus Uji	MA_01_00
Nama Kasus Uji	Kasus Uji Login Alternatif 1: Jika data <i>username</i> dan <i>password</i> kosong
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol Login pada halaman <i>home</i> 2. Aktor tidak mengisi <i>username</i> dan <i>password</i> 3. Aktor mengklik tombol login.
Hasil yang diharapkan	Sistem akan menampilkan pesan “data tidak boleh kosong!”
Hasil	Sistem akan menampilkan pesan “data tidak boleh kosong!”
Status	Valid

6.2.2 Pengujian *Logout*

Pengujian sitem untuk *logout* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_02_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Logout

Tabel 0.10 Kasus Uji Berhasil Logout

Nomor Kasus Uji	MA_02_00
Nama Kasus Uji	Login
Prosedur	1. Aktor mengklik tombol Logout pada halaman <i>dashboard</i>
Hasil yang diharapkan	Sistem menghapus <i>SESSION</i> pengguna dan menampilkan halaman login
Hasil	Sistem menghapus <i>SESSION</i> pengguna dan menampilkan halaman login
Status	Valid

6.2.3 Pengujian *Register*

Pengujian sitem untuk *register* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_03_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Register

Tabel 0.11 Kasus Uji Berhasil Register

Nomor Kasus Uji	MA_03_00
Nama Kasus Uji	Register
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol Register pada halaman <i>home</i> 2. Aktor mengisi Nama Pengguna, Kata Sandi, Email, Tipe Pelatih, Jenis Kelamin, dan mencentang term of policy. 3. Aktor menekan tombol Register.
Hasil yang diharapkan	Sistem menampilkan pesan "Berhasil melakukan registrasi" dan menampilkan halaman login
Hasil	Sistem menampilkan pesan "Berhasil melakukan registrasi" dan menampilkan halaman login
Status	Valid

2. Kasus Uji Register Alternatif 1: Data username atau email telah digunakan sebelumnya

Tabel 0.12 Kasus Uji Register Alternatif 1

Nomor Kasus Uji	MA_03_00
Nama Kasus Uji	Kasus Uji Register Alternatif 1: Data Nama Pengguna, Kata Sandi, dan Email tidak valid
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol Register pada halaman <i>home</i> 2. Aktor mengisi Nama Pengguna, Kata Sandi, dan Email dengan salah. 3. Aktor menekan tombol Register.
Hasil yang diharapkan	Sistem menampilkan pesan <i>error</i> dan menampilkan halaman register
Hasil	Sistem menampilkan pesan <i>error</i> dan menampilkan halaman register

Status	Valid
--------	-------

6.2.4 Pengujian *Reset password*

Pengujian sitem untuk *reset password* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_04_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil *Reset Password*

Tabel 0.13 Kasus Uji Berhasil *Reset Password*

Nomor Kasus Uji	MA_04_00
Nama Kasus Uji	Reset Password
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol Reset Password pada halaman <i>Login</i> 2. Aktor mengisi Email dengan benar. 3. Aktor menekan tombol <i>Reset Password</i>.
Hasil yang diharapkan	Sistem menampilkan pesan "Berhasil mengirimkan link reset password ke email anda." dan menampilkan halaman login.
Hasil	Sistem menampilkan pesan "Berhasil mengirimkan link reset password ke email anda." dan menampilkan halaman login.
Status	Valid

2. Kasus Uji Reset Password Alternatif 1: Jika email belum terdaftar kedalam sistem

Tabel 0.14 Kasus Uji Reset Password Alternatif 1

Nomor Kasus Uji	MA_04_00
Nama Kasus Uji	Kasus Uji Reset Password Alternatif 1: Email tidak valid
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengklik tombol <i>reset password</i> pada halaman <i>register</i> 2. Aktor mengisi Email dengan salah. 3. Aktor menekan tombol <i>Reset Password</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>error</i> dan menampilkan halaman <i>reset password</i>
Hasil	Sistem menampilkan pesan <i>error</i> dan menampilkan halaman <i>reset password</i>
Status	Valid

6.2.5 Pengujian Melihat Daftar Top Tim

Pengujian sitem untuk Melihat Daftar Top Tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_05_00 dan MA_05_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Top Tim

Tabel 0.15 Kasus Uji Berhasil Melihat Daftar Top Atlet

Nomor Kasus Uji	MA_05_00 dan MA_05_01
Nama Kasus Uji	Melihat daftar top tim
Prosedur	1. Aktor menekan tombol Top Tim pada halaman awal
Hasil yang diharapkan	Sistem menampilkan halaman top tim.
Hasil	Sistem menampilkan halaman top tim.
Status	Valid

6.2.6 Pengujian Melihat Daftar Top Atlet

Pengujian sitem untuk Melihat Daftar Top Atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_06_00 dan MA_06_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Top Atlet

Tabel 0.16 Kasus Uji Berhasil Melihat Daftar Top Atlet

Nomor Kasus Uji	MA_06_00 dan MA_06_01
Nama Kasus Uji	Melihat daftar top atlet
Prosedur	1. Aktor menekan tombol Top Atlet pada halaman awal
Hasil yang diharapkan	Sistem menampilkan halaman top atlet.
Hasil	Sistem menampilkan halaman top atlet.
Status	Valid

6.2.7 Pengujian Melihat Profile Pelatih

Pengujian sitem untuk Melihat Profile Pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_07_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Profile Pelatih

Tabel 0.17 Kasus Uji Berhasil Melihat Profile Pelatih

Nomor Kasus Uji	MA_07_00
------------------------	----------

Nama Kasus Uji	Melihat profile pelatih
Prosedur	1. Aktor menekan tombol profile pada menu.
Hasil yang diharapkan	Sistem menampilkan halaman profile pelatih
Hasil	Sistem menampilkan halaman profile pelatih
Status	Valid

6.2.8 Pengujian Melihat Daftar *Achievement* Pelatih

Pengujian sitem untuk melihat daftar achievement pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_08_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar *Achievement* Pelatih

Tabel 0.18 Kasus Uji Berhasil Melihat Daftar *Achievement* Pelatih

Nomor Kasus Uji	MA_08_00
Nama Kasus Uji	Melihat daftar <i>achievement</i> Pelatih
Prosedur	1. Aktor masuk ke halaman detail tim 2. Aktor memilih tab atlet dan menekan tombol lihat atlet
Hasil yang diharapkan	Sistem menampilkan daftar <i>achievement</i> atlet
Hasil	Sistem menampilkan daftar <i>achievement</i> atlet
Status	Valid

6.2.9 Pengujian Menambahkan *Achievement* Pelatih

Pengujian sitem untuk menambahkan achievement pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_09_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan *Achievement* Pelatih

Tabel 0.19 Kasus Uji Berhasil Menambahkan *Achievement* Pelatih

Nomor Kasus Uji	MA_09_00
Nama Kasus Uji	Menambahkan <i>Achievement</i> Pelatih
Prosedur	1. Aktor masuk kehalaman profile pelatih 2. Aktor menekan tombol <i>add achievement</i> . 3. Aktor mengisi form tambah achievement berupa name, description, image, level, date, coach dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> pelatih

Hasil	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> pelatih
Status	Valid

6.2.10 Pengujian Melihat Detail *Achievement* Pelatih

Pengujian sitem untuk melihat detail *achievement* pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_10_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail *Achievement* Pelatih

Tabel 0.20 Kasus Uji Berhasil Melihat Detail *Achievement* Pelatih

Nomor Kasus Uji	MA_10_00
Nama Kasus Uji	Melihat Detail <i>Achievement</i> Pelatih
Prosedur	1. Aktor masuk kehalaman profile pelatih 2. Aktor menekan tombol lihat pada tab <i>achievement</i>
Hasil yang diharapkan	Halaman detail <i>achievement</i> pelatih berhasil ditampilkan
Hasil	Halaman detail <i>achievement</i> pelatih berhasil ditampilkan
Status	Valid

6.2.11 Pengujian Mengubah *Achievement* Pelatih

Pengujian sitem untuk mengubah *achievement* pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_11_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah *Achievement* Pelatih

Tabel 0.21 Kasus Uji Berhasil Mengubah *Achievement* Pelatih

Nomor Kasus Uji	MA_11_00
Nama Kasus Uji	Mengubah <i>Achievement</i> Pelatih
Prosedur	1. Aktor masuk kehalaman detail <i>achievement</i> pelatih 2. Aktor menekan tombol edit <i>achievement</i> . 3. Aktor mengisi form mengubah <i>achievement</i> pelatih berupa name, description, image, level, date, coach dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> pelatih

Hasil	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> pelatih
Status	Valid

6.2.12 Pengujian Menghapus *Achievement* Pelatih

Pengujian sitem untuk menghapus *achievement* pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_12_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus *Achievement* Pelatih

Tabel 0.22 Kasus Uji Berhasil Menghapus *Achievement* Pelatih

Nomor Kasus Uji	MA_12_00
Nama Kasus Uji	Menghapus <i>Achievement</i> Pelatih
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> pelatih 2. Aktor menekan tombol <i>delete achievement</i>. 3. Aktor menekan tombol "Yes".
Hasil yang diharapkan	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Hasil	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Status	Valid

2. Kasus Uji Menghapus *Achievement* Pelatih Alternatif 1: Jika aktor menekan tombol cancel

Tabel 0.23 Kasus Uji Menghapus *Achievement* Pelatih Alternatif 1

Nomor Kasus Uji	MA_12_00
Nama Kasus Uji	Menghapus <i>Achievement</i> Pelatih Alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> pelatih 2. Aktor menekan tombol <i>delete achievement</i>. 3. Aktor menekan tombol "Cancel".
Hasil yang diharapkan	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> pelatih dan menampilkan halaman detail <i>achievement</i> pelatih

Hasil	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> pelatih dan menampilkan halaman detail <i>achievement</i> pelatih
Status	Valid

6.2.13 Pengujian Mengubah Profile Pelatih

Pengujian sitem untuk mengubah profile pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_13_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah Profile Pelatih

Tabel 0.24 Kasus Uji Berhasil Mengubah Profile Pelatih

Nomor Kasus Uji	MA_13_00 dan MA_13_01
Nama Kasus Uji	Mengubah profile pengguna
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman profile pelatih 2. Aktor menekan tombol edit profile 3. Aktor mengubah Username, Email, Password, Gender, Birth of Date, Address, Phone, dan Avatar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan profile pelatih
Hasil	Sistem berhasil diperbarui dan menampilkan profile pelatih
Status	Valid

2. Kasus Uji Mengubah Profile Pelatih Alternatif 1: Data *username* telah digunakan atau email telah digunakan atau phone telah digunakan

Tabel 0.25 Kasus Uji Mengubah Profile Pelatih Alternatif 1

Nomor Kasus Uji	MA_13_00 dan MA_13_01
Nama Kasus Uji	Mengubah profile pelatih alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman profile pelatih 2. Aktor menekan tombol edit profile pada halaman profile 3. Aktor mengubah data <i>avatar</i>, <i>fullname</i>, <i>username</i> yang sudah terdaftar pada

	sistem, email, password, <i>gender</i> , <i>birth of date</i> , <i>address</i> , atau <i>phone</i> dengan salah. 4. Aktor menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan error dan menampilkan halaman profile pelatih
Hasil	Sistem menampilkan pesan error dan menampilkan halaman profile pelatih
Status	Valid

6.2.14 Pengujian Melihat Detail Monitoring

Pengujian sitem untuk melihat detail monitoring dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_14_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail Monitoring

Tabel 0.26 Kasus Uji Berhasil Melihat Detail Monitoring

Nomor Kasus Uji	MA_14_00
Nama Kasus Uji	Melihat Detail Monitoring
Prosedur	1. Aktor menekan tombol monitor pada menu
Hasil yang diharapkan	Sistem menampilkan halaman daftar detail monitoring
Hasil	Sistem menampilkan halaman daftar detail monitoring
Status	Valid

6.2.15 Pengujian Menambahkan Monitoring

Pengujian sitem untuk menambahkan monitoring dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_15_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan Monitoring

Tabel 0.27 Kasus Uji Berhasil Menambahkan Monitoring

Nomor Kasus Uji	MA_15_00
Nama Kasus Uji	Menambahkan Monitoring
Prosedur	1. Aktor masuk ke halaman detail monitoring 2. Aktor menekan tombol add activity.

	3. Aktor mengisi date, place, memilih type, memilih status dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan detail monitoring.
Hasil	Sistem berhasil ditambahkan, menampilkan pesan berhasil, dan menampilkan detail monitoring.
Status	Valid

6.2.16 Pengujian Melihat daftar atlet hasil monitoring aktifitas exercise

Pengujian sitem untuk melihat daftar atlet hasil monitoring aktifitas exercise dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_16_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas exercise

Tabel 0.28 Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas exercise

Nama Kasus Uji	MA_16_00
Prosedur	1. Aktor menekan <i>home</i> pada menu
Hasil yang diharapkan	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas exercise
Hasil	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas exercise
Status	Valid

6.2.17 Pengujian Melihat daftar atlet hasil monitoring aktifitas sparing

Pengujian sitem untuk melihat daftar atlet hasil monitoring aktifitas sparing dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_17_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas sparing

Tabel 0.29 Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas sparing

Nama Kasus Uji	MA_17_00
Prosedur	1. Aktor menekan <i>home</i> pada menu
Hasil yang diharapkan	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas sparing

Hasil	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas sparing
Status	Valid

6.2.18 Pengujian Melihat daftar atlet hasil monitoring aktifitas championship

Pengujian sitem untuk melihat daftar atlet hasil monitoring aktifitas *championship* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_18_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas championship

Tabel 0.30 Kasus Uji Berhasil Melihat daftar atlet hasil monitoring aktifitas championship

Nama Kasus Uji	MA_18_00
Prosedur	1. Aktor menekan <i>home</i> pada menu
Hasil yang diharapkan	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas championship
Hasil	Sistem menampilkan melihat daftar atlet hasil monitoring aktifitas championship
Status	Valid

6.2.19 Pengujian Melihat Daftar Rekomendasi Atlet

Pengujian sitem untuk melihat daftar rekomendasi atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_19_00 dan MA_19_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Rekomendasi Atlet

Tabel 0.31 Kasus Uji Berhasil Melihat Daftar Rekomendasi Atlet

Nama Kasus Uji	MA_19_00 dan MA_19_01
Nama Kasus Uji	Melihat Daftar Rekomendasi Atlet
Prosedur	1. Aktor menekan <i>home</i> pada menu.
Hasil yang diharapkan	Sistem menampilkan daftar rekomendasi atlet pada halaman <i>home</i> .
Hasil	Sistem menampilkan daftar rekomendasi atlet pada halaman <i>home</i> .
Status	Valid

6.2.20 Pengujian Melihat Daftar Tim

Pengujian sitem untuk melihat daftar tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_20_00 dan MA_20_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Tim

Tabel 0.32 Kasus Uji Berhasil Melihat Daftar Tim

Nama Kasus Uji	MA_20_00 dan MA_20_01
Nama Kasus Uji	Melihat Daftar Tim
Prosedur	1. Aktor menekan team pada header <i>dashboard</i>
Hasil yang diharapkan	Sistem menampilkan daftar tim
Hasil	Sistem menampilkan daftar tim
Status	Valid

6.2.21 Pengujian Melihat Detail Tim

Pengujian sitem untuk melihat detail tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_21_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail Tim

Tabel 0.33 Kasus Uji Berhasil Melihat Detail Tim

Nama Kasus Uji	MA_21_00
Nama Kasus Uji	Melihat Daftar Tim
Prosedur	1. Aktor menekan team pada menu
Hasil yang diharapkan	Sistem menampilkan halaman detail tim.
Hasil	Sistem menampilkan halaman detail tim.
Status	Valid

2. Kasus Uji Melihat Detail Tim Alternatif 1: Jika parameter where null

Tabel 0.34 Kasus Uji Berhasil Melihat Detail Tim

Nama Kasus Uji	MA_21_00
Nama Kasus Uji	Melihat Daftar Tim Alternatif 1
Prosedur	1. Aktor menekan team pada menu 2. Aktor mengganti team_id pada url

Hasil yang diharapkan	Sistem akan menampilkan halaman 404 notfound
Hasil	Sistem akan menampilkan halaman 404 notfound
Status	Valid

6.2.22 Pengujian Menambahkan Tim

Pengujian sitem untuk menambahkan tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_22_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan Tim

Tabel 0.35 Kasus Uji Berhasil Menambahkan Tim

Nama Kasus Uji	MA_22_00
Nama Kasus Uji	Menambahkan tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail tim 2. Aktor menekan tombol add team 3. Aktor mengisi form add team berupa name, description, avatar, address, city, province, coach_id dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar tim.
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar tim.
Status	Valid

2. Kasus Uji Menambahkan Tim Alternatif 1: Jika name kurang dari 2 character atau lebih dari 191 character, dan / atau city lebih dari 191 character , dan/atau province lebih dari 191 character

Tabel 0.36 Kasus Uji Berhasil Menambahkan Tim

Nama Kasus Uji	MA_22_00
Nama Kasus Uji	Melihat menambahkan tim alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail tim 2. Aktor menekan tombol add team 3. Aktor mengisi form add team berupa name = kurang dari 2 character,

	description, avatar, address, city = lebih dari 191 <i>character</i> , province == lebih dari 191 <i>character</i> , coach_id dan menekan tombol submit.
Hasil yang diharapkan	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim
Hasil	Sistem akan menampilkan pesan error dan menampilkan halaman detail tim
Status	Valid

6.2.23 Pengujian Mengubah Tim

Pengujian sitem untuk mengubah tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_23_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah Tim

Tabel 0.37 Kasus Uji Berhasil Mengubah Tim

Nama Kasus Uji	MA_23_00
Nama Kasus Uji	Mengubah tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail tim 2. Aktor menekan tombol edit team 3. Aktor mengisi form edit team berupa name, description, avatar, address, city, province, coach_id secara benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail tim.
Hasil	Sistem berhasil diperbarui dan menampilkan detail tim.
Status	Valid

6.2.24 Pengujian Menghapus Tim

Pengujian sitem untuk menghapus tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_24_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus Tim

Tabel 0.38 Kasus Uji Berhasil Menghapus Tim

Nama Kasus Uji	MA_24_00
-----------------------	----------

Nama Kasus Uji	Menghapus tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail tim 2. Aktor menekan tombol delete team 3. Aktor menekan tombol hapus
Hasil yang diharapkan	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>
Hasil	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>
Status	Valid

6.2.25 Pengujian Melihat Daftar *Achievement* Tim

Pengujian sitem untuk melihat daftar *achievement* tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_25_00 dan MA_25_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar *Achievement* Tim

Tabel 0.39 Kasus Uji Berhasil Melihat Daftar *Achievement* Tim

Nomor Kasus Uji	MA_25_00 dan MA_25_01
Nama Kasus Uji	Melihat daftar <i>achievement</i> tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan menu team pada menu 2. Aktor menekan tab <i>achievement</i>
Hasil yang diharapkan	Sistem menampilkan daftar <i>achievement</i> tim
Hasil	Sistem menampilkan daftar <i>achievement</i> tim
Status	Valid

6.2.26 Pengujian Melihat Detail *Achievement* Tim

Pengujian sitem untuk melihat detail *achievement* tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_26_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail *Achievement* Tim

Tabel 0.40 Kasus Uji Berhasil Melihat Detail *Achievement* Tim

Nomor Kasus Uji	MA_26_00
Nama Kasus Uji	Melihat Detail <i>Achievement</i> Pelatih
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail tim 2. Aktor menekan tombol lihat pada tab <i>achievement</i>

Hasil yang diharapkan	Halaman detail <i>achievement</i> tim berhasil ditampilkan
Hasil	Halaman detail <i>achievement</i> tim berhasil ditampilkan
Status	Valid

6.2.27 Pengujian Menambahkan *Achievement* Tim

Pengujian sitem untuk menambahkan *achievement* tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_27_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan *Achievement* Tim

Tabel 0.41 Kasus Uji Berhasil Menambahkan *Achievement* Tim

Nomor Kasus Uji	MA_27_00
Nama Kasus Uji	Menambahkan <i>Achievement</i> Tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail tim 2. Aktor menekan tombol <i>add achievement</i>. 3. Aktor mengisi form tambah <i>achievement</i> berupa name, description, image, level, date, team dengan benar dan menekan tombol submit
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> tim
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> tim
Status	Valid

6.2.28 Pengujian Mengubah *Achievement* Tim

Pengujian sitem untuk mengubah *achievement* tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_28_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah *Achievement* Tim

Tabel 0.42 Kasus Uji Berhasil Mengubah *Achievement* Tim

Nomor Kasus Uji	MA_28_00
Nama Kasus Uji	Mengubah <i>Achievement</i> Tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> tim 2. Aktor menekan tombol edit <i>achievement</i>.

	3. Aktor mengisi form mengubah <i>achievement</i> tim berupa name, description, image, level, date, team dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> tim
Hasil	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> tim
Status	Valid

6.2.29 Pengujian Menghapus *Achievement* Tim

Pengujian sitem untuk menghapus *achievement* tim dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_29_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus *Achievement* Tim

Tabel 0.43 Kasus Uji Berhasil Menghapus *Achievement* Tim

Nomor Kasus Uji	MA_29_00
Nama Kasus Uji	Menghapus <i>Achievement</i> Tim
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> tim 2. Aktor menekan tombol <i>delete achievement</i>. 3. Aktor menekan tombol "Yes".
Hasil yang diharapkan	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Hasil	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Status	Valid

2. Kasus Uji Menghapus *Achievement* Tim Alternatif 1: Jika aktor menekan tombol cancel

Tabel 0.44 Kasus Uji Menghapus *Achievement* Tim Alternatif 1

Nomor Kasus Uji	MA_29_00
Nama Kasus Uji	Menghapus <i>Achievement</i> Tim Alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> pelatih

	2. Aktor menekan tombol <i>delete achievement</i> . 3. Aktor menekan tombol "Cancel".
Hasil yang diharapkan	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> tim dan menampilkan halaman detail <i>achievement</i> tim
Hasil	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> tim dan menampilkan halaman detail <i>achievement</i> tim
Status	Valid

6.2.30 Pengujian Melihat Daftar Atlet

Pengujian sitem untuk melihat daftar atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_30_00 dan MA_30_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Atlet

Tabel 0.45 Kasus Uji Berhasil Melihat Daftar Atlet

Nama Kasus Uji	MA_30_00 dan MA_30_01
Nama Kasus Uji	Melihat Daftar Atlet
Prosedur	1. Aktor menekan team pada menu
Hasil yang diharapkan	Sistem menampilkan daftar atlet
Hasil	Sistem menampilkan daftar atlet
Status	Valid

6.2.31 Pengujian Melihat Detail Atlet

Pengujian sitem untuk melihat detail atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_31_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail Atlet

Tabel 0.46 Kasus Uji Berhasil Melihat Detail Atlet

Nama Kasus Uji	MA_31_00
Nama Kasus Uji	Melihat Detail Atlet
Prosedur	1. Aktor masuk ke halaman detail team 2. Aktor memilih tab athlete, memilih athlete, dan menekan tombol lihat
Hasil yang diharapkan	Sistem menampilkan halaman detail atlet.

Hasil	Sistem menampilkan halaman detail atlet.
Status	Valid

6.2.32 Pengujian Menambahkan Atlet

Pengujian sitem untuk menambahkan atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_32_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan Atlet

Tabel 0.47 Kasus Uji Berhasil Menambahkan Atlet

Nama Kasus Uji	MA_32_00
Nama Kasus Uji	Menambahkan Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail Atlet 2. Aktor menekan tombol add Atlet 3. Aktor mengisi form tambah atlet berupa team, position, name, gender, avatar, address, date of birth, phone number, player number, status dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar Atlet.
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar Atlet.
Status	Valid

6.2.33 Pengujian Mengubah Atlet

Pengujian sitem untuk mengubah atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_33_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah Atlet

Tabel 0.48 Kasus Uji Berhasil Mengubah Atlet

Nama Kasus Uji	MA_33_00
Nama Kasus Uji	Mengubah Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail Atlet 2. Aktor menekan tombol edit Atlet 3. Aktor mengisi form mengubah atlet berupa team, position, name, gender,

	avatar, address, date of birth, phone number, player number, status dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail atlet.
Hasil	Sistem berhasil diperbarui dan menampilkan detail atlet.
Status	Valid

6.2.34 Pengujian Menghapus Atlet

Pengujian sitem untuk menghapus atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_34_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus Atlet

Tabel 0.49 Kasus Uji Berhasil Menghapus Atlet

Nama Kasus Uji	MA_34_00
Nama Kasus Uji	Menghapus Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail Atlet 2. Aktor menekan tombol delete Atlet 3. Aktor menekan tombol hapus
Hasil yang diharapkan	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>
Hasil	Sistem berhasil dihapus dan menampilkan halaman <i>dashboard</i>
Status	Valid

6.2.35 Pengujian Melihat Daftar *Achievement* Atlet

Pengujian sitem untuk melihat daftar *achievement* atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_35_00 dan MA_35_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar *Achievement* Atlet

Tabel 0.50 Kasus Uji Berhasil Melihat Daftar *Achievement* Tim

Nomor Kasus Uji	MA_35_00 dan MA_35_01
Nama Kasus Uji	Melihat daftar <i>achievement</i> Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail tim 2. Aktor memilih atlet dan menekan tombol lihat atlet

	3. Aktor menekan tab <i>achievement</i>
Hasil yang diharapkan	Sistem menampilkan daftar <i>achievement</i> Atlet
Hasil	Sistem menampilkan daftar <i>achievement</i> Atlet
Status	Valid

6.2.36 Pengujian Melihat Detail *Achievement* Atlet

Pengujian sitem untuk melihat detail *achievement* atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_36_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail *Achievement* Atlet

Tabel 0.51 Kasus Uji Berhasil Melihat Detail *Achievement* Atlet

Nomor Kasus Uji	MA_36_00
Nama Kasus Uji	Melihat Detail <i>Achievement</i> Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail Atlet 2. Aktor menekan tombol lihat pada tab <i>achievement</i>
Hasil yang diharapkan	Halaman detail <i>achievement</i> Atlet berhasil ditampilkan
Hasil	Halaman detail <i>achievement</i> Atlet berhasil ditampilkan
Status	Valid

6.2.37 Pengujian Menambahkan *Achievement* Atlet

Pengujian sitem untuk menambahkan *achievement* atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_37_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan *Achievement* Tim

Tabel 0.52 Kasus Uji Berhasil Menambahkan *Achievement* Atlet

Nomor Kasus Uji	MA_37_00
Nama Kasus Uji	Menambahkan <i>Achievement</i> Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail Atlet 2. Aktor menekan tombol <i>add achievement</i>. 3. Aktor mengisi form tambah <i>achievement</i> berupa name, description, image, level, date, atlet dengan benar dan menekan tombol submit
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> Atlet

Hasil	Sistem berhasil ditambahkan dan menampilkan daftar <i>achievement</i> Atlet
Status	Valid

6.2.38 Pengujian Mengubah *Achievement* Atlet

Pengujian sitem untuk mengubah *achievement* atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_39_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah *Achievement* Atlet

Tabel 0.53 Kasus Uji Berhasil Mengubah *Achievement* Atlet

Nomor Kasus Uji	MA_38_00
Nama Kasus Uji	Mengubah <i>Achievement</i> Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> Atlet 2. Aktor menekan tombol edit <i>achievement</i>. 3. Aktor mengisi form mengubah <i>achievement</i> Atlet berupa name, description, image, level, date, atlet dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> Atlet
Hasil	Sistem berhasil diperbarui dan menampilkan detail <i>achievement</i> Atlet
Status	Valid

6.2.39 Pengujian Menghapus *Achievement* Atlet

Pengujian sitem untuk menghapus *achievement* atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_39_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus *Achievement* Atlet

Tabel 0.54 Kasus Uji Berhasil Menghapus *Achievement* Atlet

Nomor Kasus Uji	MA_39_00
Nama Kasus Uji	Menghapus <i>Achievement</i> Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> Atlet 2. Aktor menekan tombol <i>delete achievement</i>. 3. Aktor menekan tombol "Yes".

Hasil yang diharapkan	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Hasil	Sistem berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Status	Valid

2. Kasus Uji Menghapus *Achievement* Atlet Alternatif 1: Jika aktor menekan tombol cancel

Tabel 0.55 Kasus Uji Menghapus *Achievement* Atlet Alternatif 1

Nomor Kasus Uji	MA_39_00
Nama Kasus Uji	Menghapus <i>Achievement</i> Atlet Alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail <i>achievement</i> Atlet 2. Aktor menekan tombol <i>delete achievement</i>. 3. Aktor menekan tombol "Cancel".
Hasil yang diharapkan	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> tim dan menampilkan halaman detail <i>achievement</i> Atlet
Hasil	Sistem akan menutup form konfirmasi menghapus <i>achievement</i> tim dan menampilkan halaman detail <i>achievement</i> Atlet
Status	Valid

6.2.40 Pengujian Melihat Daftar Riwayat Cedera Atlet

Pengujian sitem untuk melihat daftar riwayat cedera atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_40_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Riwayat Cedera Atlet

Tabel 0.56 Kasus Uji Berhasil Melihat Daftar Riwayat Cedera Atlet

Nomor Kasus Uji	MA_40_00
Nama Kasus Uji	Melihat Daftar Riwayat Cedera Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail tim 2. Aktor menekan tab atlet, memilih atlet, dan menekan tombol lihat atlet
Hasil yang diharapkan	Sistem menampilkan daftar Riwayat Cedera atlet

Hasil	Sistem menampilkan daftar Riwayat Cedera atlet
Status	Valid

6.2.41 Pengujian Melihat Detail Riwayat Cedera Atlet

Pengujian sitem untuk melihat detail riwayat cedera atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_41_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail Riwayat Cedera Atlet

Tabel 0.57 Kasus Uji Berhasil Melihat Detail Riwayat Cedera Atlet

Nomor Kasus Uji	MA_41_00
Nama Kasus Uji	Melihat Detail Riwayat Cedera Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail Riwayat Cedera 2. Aktor memilih tab <i>history injury</i>, memilih Riwayat Cedera atlet, dan menekan tombol lihat
Hasil yang diharapkan	Sistem menampilkan halaman detail Riwayat Cedera atlet.
Hasil	Sistem menampilkan halaman detail Riwayat Cedera atlet.
Status	Valid

6.2.42 Pengujian Menambahkan Riwayat Cedera Atlet

Pengujian sitem untuk menambahkan riwayat cedera atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_42_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan Riwayat Cedera Atlet

Tabel 0.58 Kasus Uji Berhasil Menambahkan Riwayat Cedera Atlet

Nomor Kasus Uji	MA_42_00
Nama Kasus Uji	Menambahkan Riwayat Cedera Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail Atlet 2. Aktor menekan tombol add injury 3. Aktor mengisi form tambah riwayat cedera atlet berupa name, description, avatar, date, athlete dengan benar dan menekan tombol submit.

Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar riwayat cedera Atlet.
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar riwayat cedera Atlet.
Status	Valid

6.2.43 Pengujian Mengubah Riwayat Cedera Atlet

Pengujian sitem untuk mengubah riwayat cedera atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_43_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil mengubah riwayat cedera atlet

Tabel 0.59 Kasus Uji Berhasil Mengubah Riwayat Cedera Atlet

Nomor Kasus Uji	MA_43_00
Nama Kasus Uji	Mengubah riwayat cedera Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail riwayat cedera Atlet 2. Aktor menekan tombol edit injury 3. Aktor mengisi form mengubah riwayat cedera atlet berupa name, description, avatar, date, athlete dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail riwayat cedera atlet.
Hasil	Sistem berhasil diperbarui dan menampilkan detail riwayat cedera atlet.
Status	Valid

6.2.44 Pengujian Menghapus Riwayat Cedera Atlet

Pengujian sitem untuk menghapus riwayat cedera atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_44_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil menghapus riwayat cedera atlet

Tabel 0.60 Kasus Uji Berhasil Menghapus Riwayat Cedera Atlet

Nomor Kasus Uji	MA_44_00
Nama Kasus Uji	Menghapus riwayat cedera Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman detail Atlet

	2. Aktor menekan tombol delete injury 3. Aktor menekan tombol hapus
Hasil yang diharapkan	Riwayat cedera atlet berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Hasil	Riwayat cedera atlet berhasil dihapus dari sistem dan menampilkan halaman <i>dashboard</i>
Status	Valid

6.2.45 Pengujian Melihat Daftar Aktifitas Atlet

Pengujian sitem untuk melihat daftar aktifitas atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_45_00 dan MA_45_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Aktifitas Atlet

Tabel 0.61 Kasus Uji Berhasil Melihat Daftar Aktifitas Atlet

Nomor Kasus Uji	MA_45_00 dan MA_45_01
Nama Kasus Uji	Melihat Daftar Aktifitas Atlet
Prosedur	1. Aktor masuk ke halaman detail tim 2. Aktor menekan tombol lihat detail atlet
Hasil yang diharapkan	Sistem menampilkan daftar aktifitas atlet
Hasil	Sistem menampilkan daftar aktifitas atlet
Status	Valid

6.2.46 Pengujian Melihat Detail Parameter Penilaian

Pengujian sitem untuk melihat detail parameter penilaian dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_46_00 dan MA_46_01. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil melihat detail parameter penilaian

Tabel 0.62 Kasus Uji Berhasil Melihat Detail Parameter Penilaian

Nomor Kasus Uji	MA_46_00 dan MA_46_01
Nama Kasus Uji	Melihat detail parameter penilaian
Prosedur	1. Aktor menekan tombol parameters pada menu
Hasil yang diharapkan	Sistem menampilkan halaman detail parameter penilaian

Hasil	Sistem menampilkan halaman detail parameter penilaian
Status	Valid

6.2.47 Pengujian Melihat Parameter Tetap Penilaian Atlet

Pengujian sitem untuk melihat parameter tetap penilaian atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_47_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Parameter Tetap Penilaian Atlet

Tabel 0.63 Kasus Uji Berhasil Melihat Detail Parameter Penilaian

Nomor Kasus Uji	MA_47_00
Nama Kasus Uji	Melihat Parameter Tetap Penilaian Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman meliaht detail parameter penilaian 2. Aktor menekan menekan tab <i>fixed parameter</i>.
Hasil yang diharapkan	Sistem menampilkan daftar parameter tetap penilaian atlet
Hasil	Sistem menampilkan daftar parameter tetap penilaian atlet
Status	Valid

6.2.48 Pengujian Melihat Daftar Parameter Tidak Tetap Penilaian Atlet

Pengujian sitem untuk melihat parameter tidak tetap penilaian atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_48_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Atlet

Tabel 0.64 Kasus Uji Berhasil Melihat Daftar Parameter Tidak Tetap Penilaian Atlet

Nomor Kasus Uji	MA_48_00
Nama Kasus Uji	Melihat Daftar Parameter Tidak Tetap Penilaian Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan menekan tab <i>your parameter</i>.

Hasil yang diharapkan	Sistem menampilkan daftar parameter tetap penilaian atlet
Hasil	Sistem menampilkan daftar parameter tetap penilaian atlet
Status	Valid

6.2.49 Pengujian Menambahkan Parameter Tidak Tetap Penilaian Atlet

Pengujian sitem untuk menambahkan parameter tidak tetap penilaian atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_49_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan parameter tidak tetap penilaian atlet

Tabel 0.65 Kasus Uji Berhasil Menambahkan Parameter Tidak Tetap Penilaian Atlet

Nomor Kasus Uji	MA_49_00
Nama Kasus Uji	Menambahkan parameter tidak tetap penilaian atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan menekan tab <i>your parameter</i>. 3. Aktor menekan tombol tambah 4. Aktor mengisi form parameter tidak tetap penilaian atlet berupa name, short name, type dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar parameter tidak tetap penilaian atlet
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar parameter tidak tetap penilaian atlet
Status	Valid

6.2.50 Pengujian Mengubah Parameter Tidak Tetap Penilaian Atlet

Pengujian sitem untuk mengubah parameter tidak tetap penilaian atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_50_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil mengubah parameter tidak tetap penilaian atlet

Tabel 0.66 Kasus Uji Berhasil Mengubah Parameter Tidak tetap Penilaian Atlet

Nomor Kasus Uji	MA_50_00
Nama Kasus Uji	Mengubah parameter tidak tetap penilaian atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan menekan tab <i>your parameter</i>. 3. Aktor memilih parameter, dan menekan tombol edit 4. Aktor mengisi form parameter tidak tetap penilaian atlet berupa name, short name, type dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan daftar parameter tidak tetap penilaian atlet
Hasil	Sistem berhasil diperbarui dan menampilkan daftar parameter tidak tetap penilaian atlet
Status	Valid

6.2.51 Pengujian Menghapus Parameter Tidak Tetap Penilaian Atlet

Pengujian sitem untuk menghapus parameter tidak tetap penilaian atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_51_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus parameter tidak tetap penilaian atlet

Tabel 0.67 Kasus Uji Berhasil Menghapus Parameter Tidak Tetap Penilaian Atlet

Nomor Kasus Uji	MA_51_00
Nama Kasus Uji	Menghapus parameter tidak tetap penilaian atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan menekan tab <i>your parameter</i>. 3. Aktor memilih parameter, dan menekan tombol delete 4. Aktor menekan tombol "Yes"

Hasil yang diharapkan	Sistem berhasil dihapus dan menampilkan daftar parameter tidak tetap penilaian atlet
Hasil	Sistem berhasil dihapus dan menampilkan daftar parameter tidak tetap penilaian atlet
Status	Valid

2. Kasus Uji Menghapus parameter tidak tetap penilaian atlet alternatif 1: Jika aktor menekan tombol Cancel

Tabel 0.68 Kasus Uji Menghapus Parameter Tidak Tetap Penilaian Atlet Alternatif 1

Nomor Kasus Uji	MA_51_00
Nama Kasus Uji	Menghapus parameter tidak tetap penilaian atlet alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan menekan tab <i>your parameter</i>. 3. Aktor memilih parameter, dan menekan tombol delete 4. Aktor menekan tombol "Cancel"
Hasil yang diharapkan	Sistem akan menutup form konfirmasi menghapus parameter tidak tetap penilaian atlet dan menampilkan daftar parameter tidak tetap penilaian atlet
Hasil	Sistem akan menutup form konfirmasi menghapus parameter tidak tetap penilaian atlet dan menampilkan daftar parameter tidak tetap penilaian atlet
Status	Valid

6.2.52 Pengujian Mengubah Persentase Tipe Parameter Penilaian

Pengujian sitem untuk mengubah persentase tipe parameter penilaian dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_52_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah Persentase Tipe Parameter Penilaian

Tabel 0.69 Kasus Uji Berhasil Mengubah Persentase Tipe Parameter Penilaian

Nomor Kasus Uji	MA_52_00
------------------------	----------

Nama Kasus Uji	Mengubah Persentase Tipe Parameter Penilaian
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan tombol edit percentage. 3. Aktor mengubah data persentase parameter penilaian dengan benar, dan menekan tombol submit
Hasil yang diharapkan	Sistem berhasil mengubah dan menampilkan detail parameter penilaian
Hasil	Sistem berhasil mengubah dan menampilkan detail parameter penilaian
Status	Valid

2. Kasus Uji Berhasil Mengubah Persentase Tipe Parameter Penilaian Alternatif 1: Jika data total persentase parameter penilaian kurang dari atau lebih dari 100%

Tabel 0.70 Kasus Uji Mengubah Persentase Tipe Parameter Penilaian Alternatif 1

Nomor Kasus Uji	MA_52_00
Nama Kasus Uji	Mengubah Persentase Tipe Parameter Penilaian Alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman melihat detail parameter penilaian 2. Aktor menekan tombol edit percentage. 3. Aktor mengubah data persentase parameter dengan total penjumlahan penilaian lebih dari 100 atau kurang dari 100, dan menekan tombol submit
Hasil yang diharapkan	Sistem akan menampilkan pesan error dan menampilkan halaman detail parameter penilaian
Hasil	Sistem akan menampilkan pesan error dan menampilkan halaman detail parameter penilaian
Status	Valid

6.2.53 Pengujian Melihat Daftar Pelatih

Pengujian sitem untuk melihat daftar pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_53_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Pelatih

Tabel 0.71 Kasus Uji Berhasil Melihat Daftar Pelatih

Nomor Kasus Uji	MA_53_00
Nama Kasus Uji	Melihat Daftar Pelatih
Prosedur	1. Aktor menekan menu halaman pelatih
Hasil yang diharapkan	Sistem menampilkan daftar pelatih.
Hasil	Sistem menampilkan daftar pelatih
Status	Valid

6.2.54 Pengujian Melihat Detail Pelatih

Pengujian sitem untuk melihat detail pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_54_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail Pelatih

Tabel 0.72 Kasus Uji Berhasil Melihat Detail Pelatih

Nomor Kasus Uji	MA_54_00
Nama Kasus Uji	Melihat Detail Pelatih
Prosedur	1. Aktor masuk ke halaman daftar pelatih 2. Aktor memilih pelatih dan menekan tombol lihat
Hasil yang diharapkan	Sistem menampilkan halaman detail pelatih.
Hasil	Sistem menampilkan halaman detail pelatih.
Status	Valid

6.2.55 Pengujian Menambahkan Pelatih

Pengujian sitem untuk menambahkan pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_55_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan Pelatih

Tabel 0.73 Kasus Uji Berhasil Menambahkan Pelatih

Nomor Kasus Uji	MA_55_00
------------------------	----------

Nama Kasus Uji	Menambahkan Pelatih
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar pelatih 2. Aktor menekan tombol add 3. Aktor mengisi form mengisi data pelatih dengan benar, dan menekan tombol submit
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar pelatih.
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar pelatih.
Status	Valid

6.2.56 Pengujian Mengubah Pelatih

Pengujian sitem untuk mengubah pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_56_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah pelatih

Tabel 0.74 Kasus Uji Berhasil Mengubah Pelatih

Nomor Kasus Uji	MA_56_00
Nama Kasus Uji	Mengubah Pelatih
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar pelatih 2. Aktor menekan tombol edit 3. Aktor mengubah data pelatih dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail pelatih.
Hasil	Sistem berhasil diperbarui dan menampilkan detail pelatih.
Status	Valid

6.2.57 Pengujian Menghapus Pelatih

Pengujian sitem untuk menghapus pelatih dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_57_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus Pelatih

Tabel 0.75 Kasus Uji Berhasil Menghapus Pelatih

Nomor Kasus Uji	MA_57_00
Nama Kasus Uji	Menghapus Pelatih
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar pelatih 2. Aktor menekan tombol delete 3. Aktor menekan tombol yes
Hasil yang diharapkan	Sistem berhasil dihapus dan menampilkan daftar pelatih
Hasil	Sistem berhasil dihapus dan menampilkan daftar pelatih
Status	Valid

6.2.58 Pengujian Melihat Daftar Cabang Olahraga

Pengujian sitem untuk melihat daftar olahraga dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_58_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Daftar Cabang Olahraga

Tabel 0.76 Kasus Uji Berhasil Melihat Daftar Cabang Olahraga

Nomor Kasus Uji	MA_58_00
Nama Kasus Uji	Melihat Daftar Cabang Olahraga
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan menu halaman cabang olahraga
Hasil yang diharapkan	Sistem menampilkan daftar cabang olahraga
Hasil	Sistem menampilkan daftar cabang olahraga
Status	Valid

6.2.59 Pengujian Melihat Detail Cabang Olahraga

Pengujian sitem untuk melihat detail olahraga dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_59_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Detail Cabang Olahraga

Tabel 0.77 Kasus Uji Berhasil Melihat Detail Cabang Olahraga

Nomor Kasus Uji	MA_59_00
------------------------	----------

Nama Kasus Uji	Melihat Detail Cabang Olahraga
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar Cabang Olahraga 2. Aktor memilih Cabang Olahraga dan menekan tombol lihat
Hasil yang diharapkan	Sistem menampilkan halaman detail Cabang Olahraga.
Hasil	Sistem menampilkan halaman detail Cabang Olahraga.
Status	Valid

6.2.60 Pengujian Menambahkan Cabang Olahraga

Pengujian sitem untuk menambahkan cabang olahraga dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_60_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menambahkan Cabang Olahraga

Tabel 0.78 Kasus Uji Berhasil Menambahkan Pelatih

Nomor Kasus Uji	MA_60_00
Nama Kasus Uji	Menambahkan Cabang Olahraga
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar pelatih 2. Aktor menekan tombol add 3. Aktor mengisi form mengisi data Cabang Olahraga dengan benar, dan menekan tombol submit
Hasil yang diharapkan	Sistem berhasil ditambahkan dan menampilkan daftar Cabang Olahraga.
Hasil	Sistem berhasil ditambahkan dan menampilkan daftar Cabang Olahraga.
Status	Valid

6.2.61 Pengujian Mengubah Cabang Olahraga

Pengujian sitem untuk mengubah cabang olahraga dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_61_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah Cabang Olahraga

Tabel 0.79 Kasus Uji Berhasil Mengubah Cabang Olahraga

Nomor Kasus Uji	MA_61_00
Nama Kasus Uji	Mengubah Cabang Olahraga
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar Cabang Olahraga 2. Aktor menekan tombol edit 3. Aktor mengubah data Cabang Olahraga dengan benar dan menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan detail Cabang Olahraga.
Hasil	Sistem berhasil diperbarui dan menampilkan detail Cabang Olahraga.
Status	Valid

6.2.62 Pengujian Menghapus Cabang Olahraga

Pengujian sitem untuk menghapus cabang olahraga dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_62_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Menghapus Cabang Olahraga

Tabel 0.80 Kasus Uji Berhasil Menghapus Cabang Olahraga

Nomor Kasus Uji	MA_62_00
Nama Kasus Uji	Menghapus Cabang Olahraga
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman daftar Cabang Olahraga 2. Aktor menekan tombol delete 3. Aktor menekan tombol yes
Hasil yang diharapkan	Sistem berhasil dihapus dan menampilkan daftar Cabang Olahraga
Hasil	Sistem berhasil dihapus dan menampilkan daftar Cabang Olahraga
Status	Valid

6.2.63 Pengujian Melihat Tampilan *Home*

Pengujian sitem untuk melihat tampilan *home* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_63_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Tampilan *Home*

Tabel 0.81 Kasus Uji Berhasil Melihat Tampilan *Home*

Nomor Kasus Uji	MA_63_00
Nama Kasus Uji	Melihat tampilan <i>home</i>
Prosedur	1. Aktor memasukkan nama website.
Hasil yang diharapkan	Sistem menampilkan halaman awal website.
Hasil	Sistem menampilkan halaman awal website
Status	Valid

6.2.64 Pengujian Melihat Tampilan *Dashboard*

Pengujian sitem untuk melihat tampilan *dashboard* dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_64_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Melihat Tampilan *Dashboard*

Tabel 0.82 Kasus Uji Berhasil Melihat Tampilan *Dashboard*

Nomor Kasus Uji	MA_64_00
Nama Kasus Uji	Melihat tampilan <i>dashboard</i>
Prosedur	1. Aktor menekan tombol <i>Home</i> saat sudah masuk ke dalam sistem
Hasil yang diharapkan	Sistem menampilkan informasi halaman awal website
Hasil	Sistem menampilkan informasi halaman awal website
Status	Valid

6.2.65 Pengujian Mengubah Score Atlet

Pengujian sitem untuk mengubah *score* atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_65_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengubah Score Atlet

Tabel 0.83 Kasus Uji Berhasil Mengubah Score Atlet

Nomor Kasus Uji	MA_65_00
------------------------	----------

Nama Kasus Uji	Mengubah Score Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail monitoring 2. Aktor mengisi data scoring berupa parameter bertipe skill, attitude, dan additional. Kemudian menekan tombol submit.
Hasil yang diharapkan	Sistem berhasil diperbarui, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.
Hasil	Sistem berhasil diperbarui, menampilkan pesan berhasil, dan menampilkan halaman detail monitoring.
Status	Valid

2. Kasus Uji Mengubah Score Atlet Alternatif 1: Jika aktor menekan tombol cancel pada halaman scoring

Tabel 0.84 Kasus Uji Mengubah Score Atlet Alternatif 1

Nomor Kasus Uji	MA_65_00
Nama Kasus Uji	Mengubah Score Atlet Alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail monitoring 2. Aktor mengisi data scoring berupa parameter bertipe skill, attitude, dan additional. Kemudian menekan tombol cancel.
Hasil yang diharapkan	Sistem akan mengagalkan masukkan pengguna dan akan menampilkan halaman scoring
Hasil	Sistem akan mengagalkan masukkan pengguna dan akan menampilkan halaman scoring
Status	Valid

3. Kasus Uji Mengubah Score Atlet Alternatif 2: Jika Tidak terdapat error pada saat memproses method input di CA_score

Tabel 0.85 Kasus Uji Mengubah Score Atlet Alternatif 2

Nomor Kasus Uji	MA_65_00
Nama Kasus Uji	Mengubah Score Atlet Alternatif 2
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail monitoring 2. Aktor mengisi data scoring berupa parameter bertipe skill, attitude, dan

	additional. Kemudian menekan tombol submit.
Hasil yang diharapkan	Sistem akan menampilkan pesan error dan menampilkan halaman scoring
Hasil	Sistem akan menampilkan pesan error dan menampilkan halaman scoring
Status	Valid

6.2.66 Pengujian Mengakhiri Monitoring Atlet

Pengujian sitem untuk mengakhiri monitoring atlet dalam melakukan pengujian terhadap kebutuhan fungsional dengan kode MA_66_00. Berikut merupakan mekanisme kasus uji yang akan dilakukan.

1. Kasus Uji Berhasil Mengakhiri Monitoring Atlet

Tabel 0.86 Kasus Uji Berhasil Mengakhiri Monitoring Atlet

Nomor Kasus Uji	MA_66_00
Nama Kasus Uji	Mengakhiri Monitoring Atlet
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail monitoring 2. Aktor menekan tombol End Activity, menampilkan formulir konfirmasi end acticity, dan menekan tombol "Yes"
Hasil yang diharapkan	Sistem berhasil diperbarui dan menampilkan halaman detail monitoring
Hasil	Sistem berhasil diperbarui dan menampilkan halaman detail monitoring
Status	Valid

2. Kasus Uji Mengakhiri Monitoring Atlet Alternatif 1: Jika aktor menekan tombol cancel pada saat menekan tombol End Activity

Tabel 0.87 Kasus Uji Mengakhiri Monitoring Atlet Alternatif 1

Nomor Kasus Uji	MA_66_00
Nama Kasus Uji	Mengakhiri Monitoring Atlet Alternatif 1
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk kehalaman detail monitoring 2. Aktor menekan tombol End Activity, menampilkan formulir konfirmasi end acticity, dan menekan tombol "cancel"
Hasil yang diharapkan	Sistem akan menutup formulir end activity dan menampilkan halaman detail monitoring.

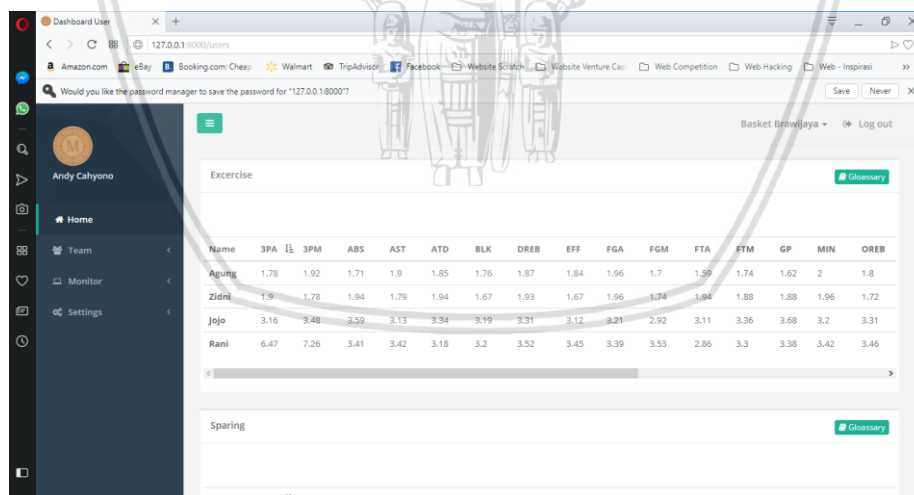
Hasil	Sistem akan menutup formulir end activity dan menampilkan halaman detail monitoring.
Status	Valid

6.2.67 Hasil Pengujian Validasi

Pengujian validasi menggunakan teknik *Scenario Base* yang didapat dari *Use Case Scenario* pada tahap analisis kebutuhan. Terdapat 66 kasus uji masing-masing kasus uji sudah menghasilkan sebuah *output* yang sesuai dengan harapan *scenario* tersebut. Maka dari itu, hasil pengujian validasi dapat dinyatakan 100% valid.

6.3 Pengujian Compatibility

Pengujian *compatibility* dilakukan dengan menjalankan sistem melalui 3 browser yang berbeda untuk melihat apakah sistem dapat berjalan dengan baik di masing-masing browser. Pada Gambar 6.4 merupakan hasil pengujian *compatibility* pada *Opera Browser*. Pada Gambar 6.5 merupakan hasil pengujian *compatibility* pada *Microsoft Edge Browser*. Pada Gambar 7.6 merupakan hasil pengujian *compatibility* pada *Chrome Browser*. Berdasarkan hasil dari ketiga kasus uji, didapatkan hasil bahwa sistem dapat berjalan dengan baik pada masing-masing *browser*.



Gambar 0.4 Pengujian Compatibility pada Opera Browser

Dashboard User | 127.0.0.1:8000/users | Basket Brawijaya | Log out

Andy Cahyono

Home | Team | Monitor | Settings

Exercise

Name	3PA	3PM	ABS	AST	ATD	BLK	DREB	EFF	FGA	FGM	FTA	FTM	GP	MIN	OREB	OT
Agung	1.78	1.92	1.71	1.9	1.85	1.76	1.87	1.84	1.96	1.7	1.59	1.74	1.62	2	1.8	1.88
Zidni	1.9	1.78	1.94	1.79	1.94	1.67	1.93	1.67	1.96	1.74	1.94	1.88	1.88	1.96	1.72	1.91
Jojo	3.16	3.48	3.59	3.13	3.34	3.19	3.31	3.12	3.21	2.92	3.11	3.36	3.68	3.2	3.31	3.28
Rani	6.47	7.26	3.41	3.42	3.18	3.2	3.52	3.45	3.39	3.53	2.86	3.3	3.38	3.42	3.46	3.14

Sparing

Name	3PA	3PM	ABS	AST	ATD	BLK	DREB	EFF	FGA	FGM	FTA	FTM	GP	MIN	OREB	OT
Agung	3.24	4.34	3.32	3.87	4.22	3.38	4.13	3.88	2.62	3.79	3.36	3.99	4.51	4	3.05	3.96

Gambar 0.5 Pengujian *Compatibility* pada Microsoft Edge Browser

Dashboard User | 127.0.0.1:8000/users | Basket Brawijaya | Log out

Andy Cahyono

Home | Team | Monitor | Settings

Exercise

Name	3PA	3PM	ABS	AST	ATD	BLK	DREB	EFF	FGA	FGM	FTA	FTM	GP	MIN	OREB	OT
Agung	1.78	1.92	1.71	1.9	1.85	1.76	1.87	1.84	1.96	1.7	1.59	1.74	1.62	2	1.8	1.88
Zidni	1.9	1.78	1.94	1.79	1.94	1.67	1.93	1.67	1.96	1.74	1.94	1.88	1.88	1.96	1.72	1.91
Jojo	3.16	3.48	3.59	3.13	3.34	3.19	3.31	3.12	3.21	2.92	3.11	3.36	3.68	3.2	3.31	3.28
Rani	6.47	7.26	3.41	3.42	3.18	3.2	3.52	3.45	3.39	3.53	2.86	3.3	3.38	3.42	3.46	3.14

Sparing

Name	3PA	3PM	ABS	AST	ATD	BLK	DREB	EFF	FGA	FGM	FTA	FTM	GP	MIN	OREB	OT
Agung	3.24	4.34	3.32	3.87	4.22	3.38	4.13	3.88	2.62	3.79	3.36	3.99	4.51	4	3.05	3.96

Gambar 0.6 Pengujian *Compatibility* pada Chrome Browser

BAB 7 KESIMPULAN

7.1 Kesimpulan

Hasil analisis yang dilakukan dalam membangun sistem monitoring dan seleksi atlet adalah membuat sistem yang memiliki fitur utama yaitu untuk menambahkan tim, melihat detail tim, dan mengubah score atlet dengan 66 kebutuhan fungsional sistem dan 1 kebutuhan non-fungsional. Aktor yang terdapat pada sistem ini ada tiga yaitu pengguna, pelatih dan atlet. Hasil analisis ini didapatkan dari studi literatur, observasi, wawancara dan validasi kesesuaian kebutuhan yang dibuat dengan keinginan client.

Perancangan yang dilakukan menggambarkan pemodelan dari fitur-fitur hasil dari analisis. Dalam tahap perancangan dibuat *sequence diagram* sebanyak 66 *sequence diagram*. Dalam tahap perancangan *class diagram*, *class diagram* yang dibuat menghasilkan 27 klas. Perancangan klas diagram ini menerapkan arsitektur MVC (*Model – View – Controller*) dimana terdapat 3 modul yaitu modul *model*, *view*, dan *controller*. Kemudian perancangan database yang menghasilkan 14 entitas yaitu *user*, *statuses*, *roles*, *position_types*, *coaches*, *sports*, *scores*, *athletes*, *teams*, *achivements*, *percentages*, *history_injuries*, dan *activities*, dan perancangan system interface yang dihasilkan sesuai dengan jumlah view.

Hasil implementasi yang dilakukan adalah menghasilkan sistem monitoring dan seleksi atlet yang memiliki fitur untuk mengelola tim, atlet, *achievement* dan riwayat cedera, dan rekomendasi atlet sesuai dengan tahapan analisis yang telah dilakukan sebelumnya. Sistem ini dibangun dengan menggunakan arsitektural Mode-View-Controller (MVC) dengan menggunakan *framework* Laravel.

Hasil pengujian yang dilakukan dengan menggunakan metode *whitebox* testing berupa pengujian unit. Hasil dari pengujian unit pada ketiga algoritme yang diuji adalah termasuk dalam range 1-10 Menurut (Guru99, 2018), hasil pengujian unit dalam range 1-10 berarti sistem yang dibangun memiliki kode struktur yang baik dan high testability. Pengujian blackbox yang dilakukan berupa validasi menghasilkan validitas sebesar 100%. Maka dapat disimpulkan aplikasi yang telah selesai dikembangkan bekerja dengan baik.

7.2 Saran

Adapun saran-saran dalam penelitian ini yaitu sebagai berikut:

1. Untuk pengembangan lebih lanjut, dapat ditambahkan metode sistem pendukung keputusan agar *score* yang di hasilkan lebih optimal.
2. Untuk pengembangan lebih lanjut, dapat ditambahkan fitur yang terhubung melalui perangkat *Internet of Things* (IoT) seperti mengukur detak jantung saat latihan dan sebagainya.

DAFTAR PUSTAKA

- Adhiasta, M. R., 2016. Perancangan Informasi Batik Solo Berbasis Bootstrap. *Jurnal Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta*, Volume 1, p. 5.
- Andriyani & Siyoperman, G., 2016. *Sistem Monitoring Peralatan Bengkel Menggunakan Metode Waterfall dengan MVC Codeigniter*. Jakarta Selatan: Universitas Tama Jagakarsa.
- Arms, W. Y., 2014. *Use Case*. [Online] Available at: <https://www.cs.cornell.edu/courses/cs5150/2014fa/slides/D2-use-cases.pdf> [Accessed 10 May 2018].
- Azdy, R. A. & SN, A., 2012. *IMPLEMENTASI SCRUM PADA PENGEMBANGAN SOFTWARE TERDISTRIBUSI*. Yogyakarta: UPN "Veteran".
- Bassil, Y., 2012. A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (iJET)*, p. 2.
- Bell, T., 2017. *Pseudo code Tutorial and Exercises*. Christchurch: University of Canterbury.
- Binarso, Y. A., Sarwoko, E. A. & Bahtiar, N., 2012. Pembangunan Sistem Informasi Alumni Berbasis Web Pada Program Studi Teknik Informatika Universitas Diponegoro. *Journal of Informatics and Technology*, Volume 1, p. 5.
- Faizin, F. R., 2015. Penentuan Pemain Inti Tim Bola Basket Menggunakan Metode AHP dan Promethee. *JTIK*.
- Guru99, 2018. *Learn Mccabe's Cyclomatic Complexity with Example*. [Online] Available at: <https://www.guru99.com/cyclomatic-complexity.html> [Accessed 14 July 2018].
- Herwin, 2004. *Keterampilan Sepakbola Dasar*. Yogyakarta: Diklat.
- IEEE-STD-610, 1991. A Compilation of IEEE Standard. *IEEE Standard Computer*.
- Juliantine, S. & Yudiana, 2010. *Belajar dan Pembelajaran Penjas*. Bandung: UPI.
- Lee, S., 2012. Unified Modeling Language (UML) for Database Systems and Computer Applications . *International Journal of Database Theory and Application*, Volume 5, p. 160.
- Miles, R. & Hamilton, K., 2006. *Learning UML 2.0*. Sebastopol: O'Reilly Media.
- Nala, 2011. *Prinsip Pelatihan Fisik Olahraga*. Denpasar: Universitas Udayana.
- Nasita, D., 2016. *Bikin Framework Sendiri dengan Teknik OOP & MVC*. Yogyakarta: Lokomedia.

Paradigm, V., 2016. *Class Diagram*. [Online] Available at: <https://www.visualparadigm.com/VPGallery/diagrams/Class.html> [Accessed 29 April 2018].

Paradigm, V., 2016. *Sequence Diagram*. [Online] Available at: <https://www.visualparadigm.com/VPGallery/diagrams/Sequence.html> [Accessed 29 4 2018].

PJ, T., 2017. *10 Advantages of Php Over Other*. [Online] Available at: <http://www.webnethosting.net/10-advantages-of-php-over-otherlanguages> [Accessed 1 May 2018].

Pressman, R. S. & Maxim, B. R., 2015. *Software Engineering A Practitioner's Approach Eighth Edition*. New York: McGraw-Hill Education.

Royce, W., 1970. Managing the Development of Large. *Proceedings of IEEE WESCON*, pp. 1-9.

Satzinger, J. W., Jackson, R. W. & Burd, S. D., 2007. Object-Oriented Analysis and Design: With the Unified Process. In: Boston: Course Technology.

Saxena, A. & Upadhyay, P., 2016. Waterfall vs. Prototype: Comparative Study of SDLC. *Imperial Journal of Interdisciplinary Research (IJIR)*, 11(6), p. 1013.

Solanki, N., Shah, D. & Shah, A., 2017. *A Survey on different Framework of PHP*. Ahmedabad: International Journal of Latest Technology in Engineering, Management & Applied Science.

Stellman, A. & Greene, J., 2005. *Applied Software*. s.l.:O'Reilly Media.

Sucipto, (2000).. *Sepakbola*. Jakarta:: Departemen Pendidikan dan Kebudayaan.

Toal, R., 2016. *Programming Paradigms*. [Online] Available at: <http://cs.lmu.edu/~ray/notes/paradigms/> [Accessed 29 4 2018].

W3C, 2016. *HTML & CSS*. [Online] Available at: <https://www.w3.org/standards/webdesign/htmlcss> [Accessed 30 April 2018].

Yudanto, A. L., 2017. Rancang Bangun Aplikasi Sistem Informasi Manajemen Laboratorium. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1.

Yu, H. R., 2014. *Design and implementation of web based on Laravel framework*. s.l.:Nankai University Binhai College, 300270.